站长百科
www.zzbaike.com
本教程由站长百科收集整理

Broadview

Java技术大系

Eclipse

精要与高级开发技术

张云涛　龚　玲　编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Broadview 技术凝聚实力·专业创新出版

Java 技术大系

Eclipse 精要与高级开发技术

Eclipse 是一个便让人着迷的开发环境，它提供的核心框架和可扩展的插件机制给广大的程序员提供了无限的想像力和创造空间。Eclipse 自身包含大量的内置功能，加上功能丰富的插件，将构成一个丰富多彩的工作环境而不仅仅是一个 IDE。

本书包含以下内容：

- Eclipse 平台体系结构
- Eclipse 的 Java 集成开发环境
- Eclipse 中的 Java 程序调试
- Eclipse 配置管理
- Eclipse 开发环境中的应用服务器 Tomcat
- Eclipse 下的 Web 应用开发
- Eclipse 插件开发技术
- Eclipse 的图形用户界面的构建
- Eclipse 下的 Struts 应用开发
- Eclipse 的版本控制及 CVS
- Eclipse 中的资源构建和构建工具 Ant
- Eclipse 环境下的 MySQL 数据库应用开发

本书不仅教你如何使用 Eclipse，而且可以使你马上投入工作！

图书分类：编程语言 > Java 技术

网上订购：www.dearbook.com.cn
第二书店 · 第一服务

责任编辑：毕 宁
封面设计：张子建
崔云霞

ISBN 7-121-00727-4

9 787121 007279 >

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

ISBN 7-121-00727-4 定价：32.00 元

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Java 技术大系

Eclipse精要与高级开发技术

张云涛　龚　玲　编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

内 容 简 介

Eclipse 是一个很让人着迷的开发环境，它提供的核心框架和可扩展的插件机制给广大的程序员提供了无限的想像力和创造空间。Eclipse 自身包含大量的内置功能，加上功能丰富的插件，将构成一个丰富多彩的工作环境而不仅仅是一个 IDE。本书内容由浅入深，先介绍了 Eclipse 平台体系结构、工作台、Java 集成开发环境、Java 程序调试；再介绍了 JUnit，Eclipse 配置管理，Eclispe 开发环境中的应用服务器 Tomcat，Eclipse 中的资源构建和构建工具 Ant；最后介绍了 Eclipse 下的 Web 应用开发、插件开发技术、Struts 应用开发和 MySQL 数据库应用开发。

本书适合于 Java 程序员、自由软件爱好者及各类软件开发人员阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Eclipse 精要与高级开发技术 / 张云涛，龚玲编著. —北京：电子工业出版社，2005.1
（Java 技术大系）
ISBN 7-121-00727-4

I. E… II. ①张… ②龚… III. 软件工具－程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字（2004）第 136450 号

责任编辑：毕 宁
印 刷：北京智力达印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
经 销：各地新华书店
开 本：787×1092 1/16 印张：19.25 字数：423 千字
印 次：2005 年 1 月第 1 次印刷
印 数：5000 册 定价：32.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

前　言

开放源代码是目前软件发展的趋势之一。开放源代码运动已经获得了越来越多的关注，并取得了很大的成功，如 Linux、Apache 等。开放源代码项目的发展趋势正由操作系统逐步转向开发工具和中间件。

目前 Eclipse 项目已经发展成为了一个庞大的 Eclipse 联盟。Eclipse.org 是软件开发工具商为加快和繁荣 Eclipse 软件开发而组织的联盟。Eclipse 开放源代码社区的目标是为了创建更好的软件开发环境以及提供集成平台，共享创建易于互操作的 Eclipse 产品利益。通过合作和共享核心集成技术，基于 Eclipse 平台上的软件开发商可致力于他们的专业领域及创建新的技术。目前已有 150 多家软件公司参加到 Eclipse 项目中，如 IBM、Borland、Red Hat、Oracle、Sybase 等。

Eclipse 平台是一个开放的、可扩展的集成开发环境（IDE），它可用于创建、集成和部署各类开发工具。Eclipse 平台包含了大量的内置功能，Eclipse 平台不仅为集成软件开发工具的开发、构造和运行提供基础模块，还提供了一整套的公共服务集，并建立了项目开发人员构建应用软件和相关组件所需的框架、基础结构和交互式工作台。

Eclipse 平台的诱人之处在于它有大量的工具开发者创建性能、功能出色的集成工具集。并且用户可以很方便地灵活使用这些工具满足其特定需求。通过 Eclipse 项目，开发和维护 Eclipse 平台及相关工具，以满足软件工具开发者及其用户的需求，从而实现 Eclipse 作为工具平台的目标。

Eclipse 软件可以集成不同的软件开发工具供应商的产品，任何开发工具厂商都可以将他们的开发工具或组件加入到 Eclipse 平台中，用户可以通过相同的接口使用不同的工具，即可在同一集成环境中同时使用不同开发商开发的软件工具。这使开发者可以在一个集成环境下使用来自多个厂商的不同工具，能够使开发人员在整个开发过程中都能进行建模、构建、测试和调试工作，软件开发者可以在 Eclipse 中选择不同的开发工具，并进行整合，从而降低开发成本，提高开发效率。开发者还可以避免因使用不兼容的工具所带来的烦琐任务，并保证了相关产品的一致性。

作者从 Eclipse 开放源代码项目建立之初即关注此项技术和相关信息，深深地被她所吸引，并在最近几年一直从事 Eclipse 技术的学习、研究、应用和推广。在推动 Eclipse 普及和应用的过程中，作者感到目前 Eclipse 社区呈蓬勃发展之势，广大 Eclipse 技术爱好者迫切希望能有一些系统、全面的 Eclipse 技术资料，尤其是中文资料。

在众多的 Eclipse 技术爱好者的鼓励之下，作者一年前即着手本书的编写。在编写过程中，Eclipse 及其相关开发技术也在不断的发展、进步之中。本书试图跟踪最新的 Eclipse 技术发展动态。

在本书编撰过程中，作者努力融合各类相关的 Eclipse 中英文材料，如 Eclipse 的联机文档、Eclipse 技术专家和爱好者的研究论文、因特网和讨论组中的各类信息，在此编者就不一一致谢，详细信息参见参考文献。此外，本书还包括作者的研究和开发经验，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

以及作者与广大的 Eclipse 技术爱好者的讨论心得等，并加以整理，其间数易其稿。然而，由于 Eclipse 是一个迅速发展的技术领域，并不断有新技术、新概念、新产品等涌现，加之作者的水平和能力所限，书中的不当之处在所难免，敬请专家和读者朋友指正。

您的任何建议和批评都是我们极为宝贵的财富。如果您能将意见和建议发往 ytzhang@mail.sjtu.edu.cn、yuntao_zhang@hotmail.com 或 lgong@mail.sjtu.edu.cn，我们将不胜感激。作者将努力和广大的 Eclipse 技术爱好者一起分享您的真知灼见，并在新版本中改进不足之处。

让我们携起手来一起推动 Eclipse 社区的发展吧！

张云涛　龚　玲

2004 年 12 月于上海交通大学

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

目　录

第 1 章　Eclipse 简介 …… 1
1.1　什么是 Eclipse …… 1
1.2　Eclipse 项目 …… 2
1.2.1　Eclipse 平台 …… 2
1.2.2　Java 开发工具 …… 3
1.2.3　插件开发环境 …… 4
1.3　Eclipse 工具项目 …… 5
1.4　Eclipse 技术项目 …… 5
1.5　Eclipse 的发展历程 …… 5
1.6　如何获取 Eclipse 程序和代码 …… 6
1.7　Eclipse 软件开发工具箱的安装和运行 …… 6
1.8　寻求帮助和报告错误 …… 10
第 2 章　Eclipse 平台体系结构 …… 11
2.1　平台基本机制 …… 11
2.2　平台运行时 …… 13
2.3　工作空间（Workspace） …… 14
2.4　工作台（Workbench） …… 15
2.5　标准小窗口工具箱和图形用户界面框架 …… 16
2.5.1　SWT …… 16
2.5.2　JFace …… 17
2.6　团队协同开发 …… 18
2.7　帮助 …… 19
2.8　外部工具 …… 20
第 3 章　工作台 …… 22
3.1　Eclipse 工作台概述 …… 22
3.1.1　透视图 …… 23
3.1.2　视图 …… 25
3.1.3　编辑器 …… 26
3.2　Eclipse 平台的菜单 …… 29
3.3　Eclipse 平台的工具栏 …… 30
3.4　创建项目 …… 31
3.4.1　使用文件菜单 …… 32

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

3.4.2 使用上下文导航菜单 …… 33
3.4.3 使用新建向导工具按钮 …… 34
3.5 资源管理 …… 34
3.5.1 资源管理概述 …… 34
3.5.2 浏览和查找资源 …… 35
3.5.3 创建文件和文件夹 …… 38
3.5.4 导入导出资源 …… 39
3.5.5 删除资源 …… 41
3.5.6 复制、重命名和移动资源 …… 42
3.5.7 资源比较 …… 42
3.6 任务及任务视图 …… 44
3.7 书签和书签视图 …… 46
3.8 定制工作台 …… 47
3.8.1 定制快速视图 …… 47
3.8.2 定制工具栏 …… 48
3.8.3 定制键绑定 …… 49
3.8.4 更改字体和颜色 …… 49
3.8.5 更改选项卡的放置 …… 50
3.8.6 定制透视图 …… 50
第 4 章 Eclipse 的 Java 集成开发环境 …… 53
4.1 验证 JRE 安装和类路径变量 …… 53
4.2 在 Eclipse 中编写一个简单的 Java 程序 …… 54
4.2.1 Java 透视图 …… 54
4.2.2 创建一个新的 Java 项目 …… 55
4.2.3 创建一个新的 Java 包 …… 55
4.2.4 创建一个新的 Java 类 …… 56
4.2.5 代码编写和代码辅助功能 …… 58
4.2.6 编译和运行程序 …… 60
4.3 Eclipse 的定制 …… 62
4.3.1 首选项 …… 63
4.3.2 定制 JRE …… 64
4.3.3 格式化 Java 代码 …… 66
4.4 其他的与 Java 编程紧密相关的透视图和视图 …… 68
4.4.1 包资源管理器视图 …… 68
4.4.2 层次结构视图 …… 68
4.4.3 控制台视图 …… 69
4.4.4 浏览透视图 …… 70

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

4.4.5 Java 类型层次结构透视图 …… 71

第 5 章 Eclipse 中的 Java 程序调试 …… 72

5.1 概述 …… 72
5.2 调试透视图简介 …… 72
5.3 常规调试技术 …… 73
5.3.1 断点 …… 73
5.3.2 调试透视图 …… 74
5.3.3 控制程序执行 …… 76
5.3.4 检查程序的执行情况 …… 78
5.4 高级调试技术 …… 78
5.4.1 表达式的计算和显示 …… 78
5.4.2 查看和修改变量和字段的值 …… 79
5.4.3 条件断点 …… 81
5.4.4 Java 异常断点 …… 81
5.4.5 监视点 …… 82
5.4.6 方法断点 …… 83
5.4.7 代码片段编辑测试窗页面 …… 84
5.4.8 远程调试 …… 86
5.5 热代码替换 …… 87

第 6 章 JUnit …… 88

6.1 单元测试简介 …… 88
6.1.1 单元测试概述 …… 88
6.1.2 JUnit 测试的作用 …… 89
6.1.3 JUnit 的获取和安装 …… 89
6.1.4 Eclipse 中 JUnit 的设置 …… 90
6.2 JUnit 测试案例 …… 90
6.2.1 在 Eclipse 中创建测试案例 …… 90
6.2.2 向测试案例中添加测试方法 …… 92
6.2.3 运行测试案例 …… 93
6.2.4 创建第二个测试案例 …… 95
6.2.5 编写测试案例的考虑事项 …… 96
6.3 测试套件 …… 97
6.3.1 在 Eclipse 中创建 JUnit 测试套件 …… 97
6.3.2 运行测试套件 …… 98
6.4 测试案例和测试套件的调试执行方式 …… 98

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 7 章　Eclipse 配置管理 …… 100
7.1　Eclipse 的结构 …… 100
7.1.1　Eclipse 目录内容 …… 100
7.1.2　功能部件的作用 …… 100
7.1.3　标识运行时平台中的功能部件 …… 101
7.2　安装 / 更新透视图和相关视图 …… 101
7.2.1　安装配置视图 …… 102
7.2.2　功能部件更新视图 …… 103
7.2.3　预览视图 …… 103
7.2.4　外部预览视图 …… 104
7.3　更新管理 …… 105
7.3.1　在 Eclipse 中安装插件 …… 105
7.3.2　在 Eclipse 中更新插件 …… 106
7.3.3　使用组更新一次安装几个功能部件 …… 107
7.4　Eclipse 中的插件配置管理 …… 108
7.4.1　检查 Eclipse 平台的当前配置 …… 108
7.4.2　保存 Eclipse 平台的当前配置 …… 110
7.4.3　恢复已保存的配置 …… 111
7.4.4　选取正在挂起的更改 …… 111
第 8 章　Eclipse 开发环境中的应用服务器 Tomcat …… 112
8.1　Tomcat 简介 …… 112
8.2　Tomcat 的安装 …… 113
8.2.1　Tomcat 的获取 …… 113
8.2.2　Tomcat 的安装和配置 …… 113
8.2.3　Tomcat 的安装测试 …… 116
8.3　Sysdeo Tomcat 插件的安装和配置 …… 117
8.4　服务器配置文件 server.xml …… 122
第 9 章　Eclipse 下的 Web 应用开发 …… 125
9.1　Servlet …… 125
9.1.1　Servlet 简介 …… 125
9.1.2　Servlet 的处理流程 …… 126
9.1.3　Java Servlet API …… 127
9.1.4　Servlet 生命周期 …… 127
9.1.5　一个简单的 Servlet 程序的示例 …… 129
9.1.6　Eclipse 中的 Servlet 应用程序的开发 …… 130
9.2　JSP …… 131

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

9.2.1 JSP 简介 …… 131
9.2.2 JSP 的实现和运行方式 …… 132
9.2.3 JSP 的访问模型 …… 133
9.2.4 JSP 文档结构和语法概述 …… 135
9.2.5 Eclipse 中的 JSP 的创建 …… 138
9.3 Web 应用程序的打包和部署 …… 140
9.3.1 部署描述符文件 web.xml …… 140
9.3.2 Web 应用的部署 …… 142
9.4 使用 Lomboz 插件开发 J2EE 应用程序 …… 143
9.4.1 Lomboz 的获取 …… 143
9.4.2 Lomboz 的安装 …… 144
9.4.3 Lomboz 在 Eclipse 中的配置 …… 144
9.4.4 Lomboz 的 JSP 应用程序开发示例 …… 145
9.4.5 Lomboz 的 Servlet 应用程序开发示例 …… 149
9.4.6 Lomboz 中的程序调试 …… 152
第 10 章 Eclipse 插件开发技术 …… 155
10.1 Eclipse 插件开发简介 …… 155
10.1.1 Eclipse 的插件结构 …… 155
10.1.2 开发 Eclipse 插件的基本步骤 …… 156
10.1.3 定义扩展点 …… 157
10.2 使用 JDT 开发 Eclipse 插件的示例 …… 158
10.3 使用 PDE 开发插件 …… 162
10.3.1 PDE 简介 …… 162
10.3.2 PDE 的配置 …… 162
10.3.3 “插件开发”透视图和相关视图 …… 163
10.3.4 使用 PDE 开发插件的简单示例 …… 164
10.3.5 插件清单编辑器 …… 167
第 11 章 Eclipse 的图形用户界面的构建 …… 171
11.1 SWT 简介 …… 171
11.2 SWT API 的结构 …… 172
11.3 小窗口 …… 173
11.3.1 小窗口应用程序的结构 …… 173
11.3.2 小窗口应用程序的系统资源管理 …… 174
11.3.3 控件 …… 175
11.3.4 定制小窗口 …… 178
11.4 SWT 的布局 …… 181

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

11.4.1 FillLayout ····· 181
11.4.2 RowLayout ····· 181
11.4.3 GridLayout ····· 182
11.4.4 FormLayout ····· 183
11.5 SWT 的事件和线程 ····· 185
11.5.1 事件 ····· 185
11.5.2 线程 ····· 186
11.6 开发 SWT 程序的示例 ····· 189
11.7 JFace 简介 ····· 193
11.8 JFace 的体系结构 ····· 194
11.9 查看器 ····· 195
11.10 JFace 操作和添加项 ····· 196
11.11 在 Eclipse 中创建 JFace 应用程序 ····· 197
第 12 章 Eclipse 下的 Struts 应用开发 ····· 200
12.1 Struts 简介 ····· 200
12.1.1 Struts 概述 ····· 200
12.1.2 Struts 的获取和安装 ····· 201
12.1.3 Struts 的配置 ····· 201
12.2 Struts 框架概述 ····· 202
12.2.1 MVC 模式 ····· 203
12.2.2 J2EE 中 MVC 模式的实现 ····· 204
12.2.3 Struts 框架 ····· 205
12.2.4 基于 Struts 框架的开发流程 ····· 208
12.3 视图及其构建 ····· 208
12.3.1 视图简介 ····· 208
12.3.2 在 Struts 中构建视图 ····· 209
12.4 控制器及其构建 ····· 209
12.4.1 控制器简介 ····· 209
12.4.2 在 Struts 中构建控制器 ····· 210
12.5 模型及其构建 ····· 211
12.6 Struts 应用的配置 ····· 212
12.6.1 国际化 Struts 应用程序 ····· 212
12.6.2 web.xml 文件 ····· 213
12.6.3 struts-config.xml 文件 ····· 214
12.7 Eclipse 中开发 Struts 应用的简单示例 ····· 215
12.7.1 创建 Struts 项目 ····· 215
12.7.2 使用“Easy Action Associated With a Form”向导 ····· 219

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

12.7.3 修改 ActionForm 类 …… 223
12.7.4 修改 Action 类 …… 226
12.7.5 修改 JSP 文件 …… 227
第 13 章 Eclipse 的版本控制及 CVS …… 229
13.1 Eclipse 的本地版本控制 …… 229
13.1.1 比较操作 …… 229
13.1.2 本地历史记录 …… 231
13.2 CVS 概述 …… 232
13.3 CVS 的基本原理 …… 233
13.4 安装和配置 CVS 资源库 …… 234
13.4.1 安装 cvsnt …… 234
13.4.2 创建 CVS 资源库 …… 235
13.5 Eclipse 中实施 CVS 系统 …… 236
13.5.1 向资源库提交项目 …… 237
13.5.2 标记资源版本 …… 241
13.5.3 从 CVS 资源库检出项目和模块 …… 242
13.5.4 更新 …… 243
13.5.5 补丁程序的创建和应用 …… 244
第 14 章 Eclipse 中的资源构建和构建工具 Ant …… 247
14.1 Ant 简介 …… 247
14.1.1 Ant 概述 …… 247
14.1.2 Ant 的获取和安装 …… 247
14.1.3 Ant 的配置 …… 248
14.2 构建文件 …… 248
14.2.1 <project> …… 250
14.2.2 <target> …… 250
14.2.3 <mkdir> …… 251
14.2.4 <jar> …… 251
14.2.5 <javac> …… 251
14.2.6 <java> …… 252
14.2.7 <delete> …… 252
14.2.8 <echo> …… 253
14.2.9 <property> …… 253
14.2.10 <buildnumber> …… 253
14.2.11 <copy> …… 253
14.3 Ant 数据类型 …… 254

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

14.3.1 argument …… 254
14.3.2 environment …… 254
14.3.3 filelist …… 255
14.3.4 patternset …… 255
14.3.5 fileset …… 255
14.3.6 filterset …… 256
14.3.7 path …… 256
14.3.8 mapper …… 256
14.4 Ant 的运行 …… 256
14.5 Eclipse 和 Ant 的集成 …… 257
14.5.1 创建 Ant 构建文件 …… 257
14.5.2 编辑 Ant 构建文件 …… 258
14.5.3 运行 Ant 构建文件 …… 258
14.5.4 保存和重用 Ant 选项 …… 259
14.5.5 Eclipse 中的 Ant 视图 …… 260
14.5.6 修改 Ant 类路径 …… 261
14.5.7 使用其他版本的 Ant …… 262
14.6 在 Eclipse 中使用 Ant 的示例 …… 263
第 15 章 Eclipse 环境下的 MySQL 等数据库应用开发 …… 265
15.1 MySQL 简介 …… 265
15.1.1 MySQL 的获取和安装 …… 265
15.1.2 MySQL 的配置 …… 265
15.2 数据库连接 …… 266
15.2.1 JDBC 对象 …… 266
15.2.2 JDBC 驱动程序的类型 …… 267
15.2.3 JDBC 的使用 …… 268
15.2.4 MySQL 数据连接池 …… 270
15.2.5 一个简单的数据库查询的例子 …… 272
附录 A 开放源代码软件 …… 273
附录 B 自由软件 …… 278
附录 C 通用公共许可证 …… 280
附录 D 通用公共许可证的法律要点和热点问题的阐释 …… 285
参考文献 …… 290

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 1 章　Eclipse 简介

开放源代码是目前软件发展的趋势之一。开放源代码运动已经获得了越来越多的关注，并取得了很大的成功，如 Linux、Apache 等。开放源代码项目的发展趋势正由操作系统逐步转向开发工具。

1.1　什么是 Eclipse

Eclipse 软件是 IBM 投入巨资开发，并捐赠给国际开放源代码团体，由新成立的开放源代码协会——Eclipse 组织进行管理的基于 Java 的开发平台软件。虽然 Eclipse 软件基于 Java，支持 JDK 和用于生成代码文档的 JavaDoc 标准，但根据 Eclipse 的体系结构，通过开发插件，它能扩展到任何语言的开发，甚至能成为图片绘制的工具。因此，Eclipse 也可支持 C，C++，C#，PL1，COBOL 等编程语言。

Eclipse 软件可以集成不同的软件开发工具供应商的产品，任何开发工具厂商都可以将他们的开发工具或组件加入到 Eclipse 平台，用户可以通过相同的接口使用不同的工具，即可在同一集成环境中同时使用不同开发商开发的软件工具。这使开发者可以在一个集成环境下使用来自多个厂商的不同工具，能够使开发人员在整个开发过程中都能进行建模、构建、测试和调试工作；软件开发者可以在 Eclipse 中选择不同的开发工具，并进行整合，从而降低开发成本，提高开发效率。开发者还可以避免因使用不兼容的工具所带来的烦琐任务，并保证了相关产品的一致性。

由于应用系统中经常需要综合多个中间件产品和不同的工具，这些产品和工具间可能存在兼容性问题。而基于 Eclipse 开发平台，可以整合不同的工具，开发者可以综合性能调节、监测、调试等处理，从而可以在较短的时间内开发出更高性能和质量的应用。因此，Eclipse 平台不仅得到了许多开发者和新兴的中小软件公司的支持，而且得到 IBM、Sun、Oracle 等大型软件公司的支持。

Eclipse 软件可以简化用于多操作系统软件工具的开发过程，插件可以为 Eclipse 可移植 API 设计程序，并在任何受支持的操作系统上不加改变地运行。基于 Eclipse 的软件在 Linux 和微软 Windows 系统上都可以运行，因此可以很方便地将所开发的软件在 Windows 操作系统和 Linux 操作系统间进行切换。Eclipse 将会不断支持更多的操作系统和编程语言。Eclipse 支持 Sun 公司的 JDK 和用于生成代码文档的 JavaDoc 标准。Eclipse 2 支持 Solaris, HP-UX, AIX 以及 QNX 操作系统，并且支持 Java 开发工具包 1.4 版本。

总而言之，Eclipse 是一个开放源代码软件开发项目，它是应用开发工具的开发环境和运行的通用平台。Eclipse 软件将成为超越 Visual Studio.Net 等开发工具包的强有力的

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

竞争产品。Eclipse 包含三个项目，即 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目。每个项目都由项目管理委员会检查并按照相应的项目章程进行管理。每个项目都可能包含相应的子项目。这些项目都使用 CPL（Common Public License）1.0 版本的许可协议。

1.2 Eclipse 项目

Eclipse 项目的主要目的是提供一个功能强大的、稳定的平台，在该平台上开发商（开发者）可以进行工具开发，并且这些工具可以高度集成。它由三个子项目组成：平台子项目、Java 开发工具（JDT-Java）子项目、插件（PDE-Plug-in）开发环境子项目。

Eclipse 平台的成功依赖于它是否能真正使各类工具开发商创建最佳的集成开发工具。但是，只有在用户能够根据其特定的需求将这些不同的工具开发商开发的工具组合在一起来工作，而无须按照工具开发商设想的方式使用，Eclipse 作为工业平台的设想才能真正实现。Eclipse 项目的任务是采用和发展 Eclipse 平台及相关工具，以满足工具开发组织及其用户的需求，从而实现将 Eclipse 作为真正的工业平台的设想。

Eclipse 项目是一个开放源代码项目，致力于让工具开发人员和工具开发商开发基础代码以创建他们自己的工具，所生成的产品通常称为工作台。

1.2.1 Eclipse 平台

Eclipse 平台是一个开放的可扩展的集成开发环境（IDE），它可用于创建、集成和部署各类开发工具。Eclipse 平台包含大量内置功能，Eclipse 平台为集成软件开发工具的开发、构造和运行提供基础模块，提供了一整套的公共服务集，并建立了项目开发人员构建应用软件和相关组件所需的框架、基础结构和交互式工作台。

Eclipse 平台的过人之处在于它有大量的工具开发者创建性能、功能出色的集成工具集。并且用户可以很方便地灵活使用这些工具满足其特定需求。通过 Eclipse 项目，开发和维护 Eclipse 平台及相关工具，以满足软件工具开发者及其用户的需求，从而实现 Eclipse 作为工具平台的目标。

Eclipse 软件开发工具箱（SDK）是由 Eclipse 项目的三个子项目的组件组合而成的单一可下载包。Eclipse 软件开发工具箱为开发者提供了功能强大的开发环境，可以高效地开发和 Eclipse 平台无缝集成的软件工具。由于 Eclipse 软件开发工具箱由 Eclipse 项目软件和其他一些开放源代码项目的第三方软件组成，因此除了 Eclipse 项目软件需要遵循通用公共许可之外，Eclipse 软件开发工具箱中的第三方组件将遵循各自的软件许可。这一点是 Eclipse 软件开发工具箱的使用者要格外注意的。

所有基于 Eclipse 的工具都在 Eclipse 平台上创建。Eclipse 平台又可进一步划分为：

- Ant ——基于 Java 的编译工具
- Compare ——通用比较工具
- Core ——核心库

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
第 1 章 本教程由站长百科收集整理

- Debug ——通用调试器
- Help ——帮助系统
- Scripting ——脚本支持
- Search ——集成的搜索工具
- SWT ——标准小窗口工具箱
- UI ——用户界面
- Update ——动态更新 / 安装服务
- VCM ——版本化和配置管理
- WebDAV ——WebDAV 集成

1.2.2 Java 开发工具

目前，Java 开发领域有大量的基于集成开发环境(IDE)的开发工具，如 IBM 的 Visual Age for Java、WebSphere Studio，Borland 的 JBuilder，以及开放源代码的 NetBeans 和 Eclipse 等。其中，Eclipse 由于其自身的各种优势，是一个重要的并不断快速发展的 Java 开发工具。

Eclipse 是一个开放源代码的项目，任何组织和个人都可以下载 Eclipse 的源代码，并且在此基础上开发自己的功能插件。因此，可以通过开发新的插件支持不同的编程语言。同时可以通过开发新的插件扩展现有插件的功能，比如在现有的 Java 开发环境中加入 Tomcat 服务器插件。由于可以不断扩展功能，而且有着统一的接口、操作和系统资源管理，这也正是 Eclipse 的优势所在。

JDT 项目给出一组插件，这些插件将各种 Java IDE 功能添加到 Eclipse 平台中。JDT 插件提供一些应用程序编程接口（API），其他的工具开发者可以使用这些 API 对这些插件进行进一步的扩展。JDT 插件可分为 JDT 核心、JDT 用户界面（UI）和 JDT 调试这几类。

1. JDT 核心

JDT 核心定义非用户界面的基础结构。它包括以下方面：

- 增量式 Java 构建器（builder）
- Java 模型，它提供浏览 Java 元素树的 API
- 代码辅助和代码选择支持
- 用于搜索、代码辅助、类型层次计算和重建的基于索引的搜索基础结构
- 支持对代码片段编辑测试窗或调试器中的代码片段进行赋值计算

Java 元素树定义了以 Java 为中心的项目视图。它表示了一组包段（package fragment）、编译单元、二进制类、类型、方法和域等 Java 元素。

因为 JDT 核心基础结构将所需的文件作为安装的一部分，所以它没有内置的 Sun Java 开发包（JDK）版本依赖。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

2. JDT 用户界面

JDT 用户界面实现特定于 Java 的工作台功能：

- 包视图
- 类型层次视图
- Java 大纲视图
- 创建 Java 元素的向导
- Java 编辑器

Java 编辑器提供下列功能：

- 关键词和句法着色
- 基于特定上下文（Java，Javadoc）的代码辅助和代码选择
- 方法级编辑
- 问题、断点和搜索匹配的旁注
- 编辑时大纲视图将自动更新
- API 帮助显示弹出窗口中选中的 Java 元素的 Javadoc 规格说明
- 导入帮助，并自动创建和组织导入声明
- 格式化代码

JDT 用户界面提供重建支持，如 Java 元素的抽取方法或安全重命名，同时还更新引用。用户可以预览（或禁止）由于重建操作而引起的单个单元的更改。

JDT 搜索支持准确搜索，如查找包、类型、方法和域的声明和引用，查找工作空间的范围、工作集或当前选择。

JDT 比较操作实现了 Java 编译单元的结构化比较，它可显示对单个 Java 方法的修改，并可以将单个的 Java 元素用本地历史中的老版本代替。

3. JDT 调试

JDT 调试实现 Java 调试功能，可使用任何遵循 JDPA 规范的 Java 虚拟机（JVM）。平台调试器提供了独立于语言的“调试模型”，在其上可实现 JDT 调试。

JDT 调试提供以下调试功能：

- 既可以按运行模式也可以按调试模式启动 JVM
- 附属于运行着的 JVM
- 在堆栈框架的上下文中进行表达式赋值
- 支持交互式的 Java 代码片段赋值计算的剪贴簿页
- 当 JVM 支持时，可进行动态类重载

1.2.3 插件开发环境

为了便于构建 Eclipse 插件，PDE 项目提供了许多视图和编辑器。使用 PDE 可以创建插件清单文件（plugin.xml）；指定插件运行时和其他所需的插件；定义扩展点，包括特定的标记；将扩展点标记与 XML 模式文件进行关联从而对扩展进行验证；在其他插

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

件扩展点创建扩展等。PDE 使得集成插件变得容易且有趣。

插件使用 Java 编写，并且为了便于插件的创建，PDE 提供了一个插件的开发和测试环境。插件开发环境（PDE）又可进一步划分为：

- PDE Core ——PDE 内核
- PDE UI ——PDE 用户界面

1.3 Eclipse 工具项目

Eclipse 工具项目的主要作用是促进 Eclipse 平台上各种工具的创建。为不同的工具开发者提供开发目标和开发重点，协调众多的工具开发商，从而最大程度地降低所开发的软件开发工具功能的交迭和重复，确保公共组件最大程度地共享和创建，并在不同类型的工具间提供无缝的互操作性。工具项目所包含的子项目由工具开发者社区和工具项目管理委员会提出、选择及开发。

1.4 Eclipse 技术项目

Eclipse 技术项目是为开放源代码软件的开发者、研究人员、学者和培训人员提供参与 Eclipse 发展的渠道的。它由三个相关的项目流程组成，即研究、孵化器和培训。研究项目探索 Eclipse 相关领域的研究课题，如编程语言、工具和开发环境。孵化器是为 Eclipse 软件添加新功能而非正式地组织的小项目。培训项目着重于培训材料、辅助教学和课件的开发。

1.5 Eclipse 的发展历程

IBM 在 2001 年 11 月为 Eclipse 开放源代码项目投资 4000 万美元，后捐献给 Eclipse 开放源代码社区。IBM 全球高级副总裁，软件集团总经理 Steve Mills 说："通过开放 Eclipse 的源代码，IBM 对于软件工具所做的，正如 Apache 对于 Web 应用服务器，Linux 对于操作系统"。

目前由 IBM 牵头，围绕着 Eclipse 项目已经发展成为一个庞大的 Eclipse 联盟。Eclipse.org 是软件开发工具商为加快和繁荣 Eclipse 软件开发而组织的联盟。目标是为了创建更好的软件开发环境以及提供集成平台，共享创建易于互操作的 Eclipse 产品利益。通过合作和共享核心集成技术，基于 Eclipse 平台的软件开发商可致力于他们的专业领域及创建新的技术。目前已有 150 多家软件公司参加到 Eclipse 项目中，如 IBM、Borland、Red Hat、Oracle、Sybase 等。

最近的 Eclipse 平台版本除可以在 Windows、Linux 上运行外，所支持的操作系统新增了 Solaris，后续版本中将支持更多的操作系统。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在 Eclipse 2.1.3 以上版本中，已经将可访问性功能部件添加至 Eclipse。可访问性功能部件帮助身体伤残或具有特殊需要的用户使用软件产品。这些功能部件可以帮助用户更有效地使用 Eclipse。例如，窗口→键盘快捷键菜单列出了常用操作的键盘快捷键。从这些细节可以看出，Eclipse 已经从自由软件爱好者的个性化产品变为通用的大众化产品。

1.6 如何获取 Eclipse 程序和代码

Eclipse 是开放源代码的项目，Eclipse 开放源代码团体的主页地址为 http://www.eclipse.org。只要遵循通用公共许可协议，任何组织或个人可以到 www.eclipse.org 站点免费下载 Eclipse 的最新版本，Eclipse 一般提供 Release，Stable Build，Integration Build 和 Nightly Build 等不同类型的下载版本。软件开发工具箱 SDK 的下载网址为：http://www.eclipse.org/downloads/index.php。

从 Web 站点下载 Eclipse 时获取的是基本组件，这些组件是免费的，为工具开发者构建自己专用的产品提供了基础。这些组件是 Eclipse 平台、Java 开发工具包和插件开发环境。

当前已有一些构建在 Eclipse 基础上的商业产品，并不断有更多的新产品出现。其中一些产品是独立的，另一些是专用的插件，用于为独立产品提供额外的或增强的功能。

由 Eclipse 项目生成的软件在通用公共许可（GPL）授权下使用。Eclipse SDK 包含其他开放源代码项目的第三方软件，因此这些第三方组件在他们自己的授权下使用。有关授权信息的细节可在 SDK 下载的根目录的 Eclipse.org 软件用户协议（notice.html）文件中找到。第三方组件的授权分别在其子目录的 about.html 文件中描述。

1.7 Eclipse 软件开发工具箱的安装和运行

Eclipse SDK（软件开发包）将 Eclipse 项目的三个子项目平台、Java 开发工具和插件开发环境生成的组件联合成单一的可下载包。应当下载最近发布的版本或最近稳定的开发版本。可在下载页下载这些版本。最近发布的版本错误较少，但最近稳定版本具有所有最近的功能部件和新特性。因此，对于初次使用者而言，使用最近的发布版更安全些，这样就不会被最近的稳定开发版中不稳定的错误所影响。在进行任何实际的工作时，应当仅使用发布版。当产生一个新的稳定版本时，新闻组会给出一个通知。

下载页上对于可供下载的各种不同平台上的不同版本的 Eclipse SDK 都有一个状态标记显示。标记为绿色的勾（√）或红色的叉（×）。绿色的√和红色的×是对版本质量的粗略估计。绿色的勾表示该版本已通过以下测试：

- 插件编译
- 通过所有自动测试
- 该版本通过一个 5~10 分钟的人工探测测试

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

如果其中之一的测试失败则该版本被标记为红色的×。标记为绿色的√的版本通常被认为已足够稳定，可为 Eclipse 开发团队所使用，但带有绿色标记的版本仍可能有错误和缺憾。

Eclipse 软件开发工具箱的安装和运行很简单，其具体步骤如下：

① 从 Eclipse 网站下载所需平台的 SDK 的压缩文件，一个比较常规的选择是选择最近的发布版，目前是 2.1.3，但稳定构建版本的最新版本是 3.0；

② 安装 JRE 或 Java SDK（如已安装，则可跳过此步）；

③ 将压缩文件解压缩到任意一个目录；

④ 在 Windows 系统上直接双击运行该目录中的可执行文件 eclipse.exe，则启动 Eclipse SDK，而在 Linux 系统上则为 eclipse。

为了以后能比较便捷地运行 Eclipse，最好在桌面上创建 Eclipse 应用程序的快捷方式。当然，用户也可以在命令提示符下将目录切换为 Eclipse 安装目录，然后运行 eclipse.exe 来获得同样的效果。

Eclipse SDK 是一个英文版的软件开发工具箱，但目前 Eclipse 已能支持多国语言，包括简体中文、繁体中文、德语、西班牙语、法语、意大利语、日语和葡萄牙语。用户如需以上语言的 Eclipse 软件开发工具箱，只需从 Eclipse 网站下载语言包 eclipse-nls-SDK-2.1.3.zip。然后将该压缩文件解压缩到 Eclipse 目录中，然后重启 Eclipse 平台即可。

用户需要注意的是：Eclipse SDK 不带 JRE。Eclipse 需要 Java 2 标准版 Java 运行时环境 JRE 的 1.3 或 1.4 版本。运行 Eclipse，必须在计算机上预先安装 JRE。因为 JRE 包含在 Java 开发工具包中，所以下载并安装 Java 开发工具包即可。

一旦安装了 Java 开发工具包，就可以使用-vm 命令行参数启动 Eclipse。例如，如果在 c:\jdk1.3 目录下安装了 Java 开发工具包，就可用下面的命令行命令启动 Eclipse：

```
eclipse.exe -vm c:\jdk1.3\bin\javaw.exe
```

用户还可以指定正在运行 Eclipse 的 JVM 的参数。例如，当在运行大项目时，如果得到内存不足的错误信息时，可增加 Java 堆的大小。

Eclipse 可执行文件及平台本身提供了开发或调试 Eclipse 各部件的许多执行选项。表 1.1 是 Eclipse 的其他一些启动参数的说明。

表 1.1 Eclipse 的启动参数

命 令 参 数	描 述
-arch architecture	定义 Eclipse 平台在其上运行的处理器体系结构。Eclipse 平台通常使用 Java os.arch 属性的常用值来计算最佳设置。如果在此处指定该项，则这是 Eclipse 平台使用的值。此处指定的值可作为 BootLoader.getOSArch()用于插件。示例值有："x86"、"sparc"、"PA-RISC" 和 "ppc"
-application applicationId	指定要运行的应用程序。应用程序由对 org.eclipse.core.runtime.applications 扩展点提供扩展的插件来声明。通常不需要此自变量。如果指定了此项，则该值会覆盖配置提供的值。如果不指定此项，则会运行 "Eclipse 工作台"

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

续表

命 令 参 数	描　　述
-boot bootJarURL	（建议不使用；用-configuration 代替；与 1.0 兼容）。Eclipse 平台的引导插件代码（boot.jar）的位置，表示为 URL。如果指定此项，则会用它来为装入 Eclipse 平台引导程序类装入器设置类路径。仅当更改 startup.jar 和 boot.jar 的相对位置时才需要它。注意，不允许使用相对 URL
-classloaderproperties [file]	如果指定的话，则使用给定位置处的类加载器属性文件来激活平台类加载器。文件自变量可以是文件路径或 URL。注意，不允许使用相对 URL
-configuration configurationFileURL	Eclipse 平台配置文件的位置，表示为 URL。配置文件确定 Eclipse 平台、可用插件集和主要功能部件的位置。注意，不允许使用相对 URL。当安装或更新 Eclipse 平台时配置文件被写至此位置
-consolelog	将 Eclipse 平台的错误日志镜像到用来运行 Eclipse 的控制台。与-debug 组合时使用很有用
-data workspacePath	要运行 Eclipse 平台的工作区的路径。工作区位置也是项目的默认位置。相对于从中启动 Eclipse 的目录来解释相对路径
-debug [optionsFile]	将平台置于调试方式。并从指定位置处的文件装入调试选项（如果指定的话）。此文件指示哪些调试点可用于插件以及是否已启用它们。如果未给出文件位置，则平台在启动 Eclipse 的目录中查找“.options”文件。URL 和文件系统路径都可作为文件位置
-dev [classpathEntries]	将平台设置为开发方式。将可选类路径条目（用逗号分隔的列表）添加至每个插件的运行时类路径。例如，当工作区包含要开发的插件时，指定-dev bin 会为每个插件项目的名为 bin 的目录添加类路径条目，允许在其中存储最新生成的类文件。除去了冗余或不存在的类路径条目
-endsplash params	用于在 Eclipse 平台启动并运行时关闭闪屏的内部选项。此选项在闪屏处理链中不同的位置有不同的语法和语义
-feature featureId	主要功能部件的标识。主要功能部件为 Eclipse 的已启动实例提供了产品个性，并确定使用产品的定制信息
-keyring keyringFilePath	磁盘上授权数据库（或“密钥环”文件）的位置。此自变量必须与-password 选项配合使用。相对于从中启动 Eclipse 的目录来解释相对路径
-nl locale	定义 Eclipse 平台在其上运行的语言环境的名称。Eclipse 平台通常自动计算最佳设置。如果在此处指定该项，则这是 Eclipse 平台使用的值。此处指定的值可作为 BootLoader.getNL()用于插件。示例值有：“en_US”和“fr_FR_EURO”
-nolazyregistrycacheloading	取消激活加载优化的平台插件注册表高速缓存。默认情况下，仅当需要时才从注册表高速缓存（可用时）中加载扩展的配置元素，以减少内存占用。此选项将在启动时强制完全装入注册表高速缓存
-noregistrycache	绕过读写内部插件注册表高速缓存文件

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

续表

命令参数	描　述
-nosplash	运行平台而不显示闪屏
-os operatingSystem	定义运行 Eclipse 平台的操作系统。Eclipse 平台通常使用 Java os.name 属性的常用值来计算最佳设置。如果在此处指定该项，则这是 Eclipse 平台使用的值。此处指定的值可作为 BootLoader.getOS() 用于插件，并用于解析插件清单文件中提及的路径中os 变量。示例值有："win32"、"linux"、"hpux"、"solaris" 和 "aix"
-password password	授权数据库的密码。与-keyring 选项配合使用
-perspective perspectiveId	启动时要在活动工作台窗口中打开透视图。如果没有指定该参数，则将打开关闭时活动的透视图
-plugincustomization propertiesFile	包含插件首选项默认设置的属性文件的位置。这些默认设置覆盖主要功能部件中指定的默认设置，相对于从中启动 eclipse 的目录来解释相对路径
-plugins pluginsFileURL	（建议不使用：用-configuration 代替，与 1.0 兼容）。指定 Eclipse 平台查找插件文件的位置，表示为 URL。该文件为属性文件格式，其中名是任意用户定义名称，值是指向 plugin.xml 文件的显式路径或指向包含插件目录路径的用逗号分隔的列表。注意，不允许使用相对 URL。如果指定此项，则此选项会导致创建适当的临时配置
-refresh	启动时执行工作区的全局刷新的选项。这将使从上次平台运行以来在文件系统中所做的任何更改都保持一致
-showlocation	用于在窗口标题栏中显示工作区的位置的选项。在发行版 2.0 中，此选项仅与-data 命令行自变量一起使用
-showsplash params	用于显示闪屏（由可执行的 Eclipse 平台启动器执行）的内部选项。此选项在闪屏处理链中不同的位置有不同的语法和语义
-vm vmPath	运行 Eclipse 平台的 "Java 运行时环境"（JRE）的位置。如果不指定此项，则 JRE 位于 jre\（它是 Eclipse 可执行文件的兄弟目录）。相对于从中启动 eclipse 的目录来解释相对路径
-ws windowSystem	定义运行 Eclipse 平台的 Windows 系统。Eclipse 平台通常使用 Java os.name 属性的常用值来计算最佳设置。如果在此处指定该项，则这是 Eclipse 平台使用的值。此处指定的值可作为 BootLoader.getWS() 用于插件，用于配置 SWT，以及用于解析插件清单文件中提及的路径中的 ws 变量。示例值有："win32"、"motif" 和 "gtk"

Eclipse 的启动参数具有强大的功能，可以定制各类所需的环境。例如，假设 Eclipse 中已经安装了语言包，而又需要运行英文环境的 Eclipse，则在启动命令中添加参数-nl "ZH_US"就可以了，即在 Eclipse 安装目录下执行：

```
eclipse.exe -nl "ZH_US"
```

另一个需要注意的是版本升级的问题。当进行版本更新时，需要删除老的版本，重

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

新进行安装，而不能将 Eclipse 压缩文件直接解压到原来的路径覆盖老版本。

由于安装 Eclipse 不修改注册表，所以当需要卸载 Eclipse 时，只需要在解压 Eclipse 的地方删除“Eclipse”目录结构。简单地删除文件就可以完全卸载 Eclipse。

1.8 寻求帮助和报告错误

Eclipse 提供了相关的帮助手册。查看手册有以下两种方式。

1. 在 Eclipse SDK 内阅读手册

通过“帮助”菜单打开“帮助内容”，然后在窗口左侧的“内容”对象树中选择相应的手册，即可阅读这些手册。

2. 在线阅读手册

在线手册既有 HTML 格式也有 PDF 格式。

有关没有在 Eclipse 技术手册中涉及的问题，用户可以将它们张贴到新闻组中。为了防止垃圾信息，用户张贴文章需要一个口令。如果没有口令，则需要先申请。请在新闻组里选择技术主题。记住，这些新闻组是公共区，因此不要在你的问题中包含任何机密信息！在加入新闻组之前，用户最好首先阅读一下 www.eclipse.org 网站中的 Eric Raymond 的“如何巧妙地问问题（How to ask questions the smart way）”一文。

对于在使用 Eclipse 的过程中，用户所发现的错误不要提交到新闻组，而应提交到 bugzilla。Eclipse 项目使用 bugzilla 作为它的错误跟踪系统。输入一个错误报告与在 Eclipse bugzilla 页上填写 Web 表格一样简单。第一次输入错误时，报告者需要通过提供一个 E-mail 地址并选择一个口令来创建一个新的 bugzilla 账户。

在输入错误报告前，应首先搜索 bugzilla 查看是否有人已经输入了类似的报告。如果找到一个涉及同样问题的错误报告，报告者只须简单地给出注释让开发者知道你也遇到了这个问题。反之，如果搜索了 bugzilla 而没有找到任何相似的问题，则可继续输入新的错误报告。有关报告错误的相关信息，可阅读在 Eclipse 错误报告页上的错误输入指南。

为了使错误报告对 Eclipse 开发者更具帮助性，报告者提交的错误报告应包括以下方面的内容。

① 所使用的 Eclipse SDK 版本。

② 运行 Eclipse 的计算机的信息（操作系统、内存和其他相关信息）。

③ Eclipse 的.log 文件的内容。这对报告一个内部错误来说尤为重要。

④ 用于运行 Eclipse 的 Java 运行时或开发工具。

⑤ 遇到错误时正在做的事的描述。

⑥ 观察到的行为与预期行为如何不同的描述。

⑦ 反复出现错误的一步步说明，使开发者可以自己重现该错误。

.log 文件在 workspaxe/.metadata 目录下。Eclipse 平台用.log 文件记录运行时错误。之所以在报告错误中包含它，主要因为它包含了在插件中发生的栈跟踪。因而报告错误时，需要包括.log 文件的相关信息。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 2 章　Eclipse 平台体系结构

Eclipse 是一种通用的工具平台。平台是在软件设计过程中，作为一个通用的中间层来连接各个功能模块，提供标准的接口，接受各个模块的请求，控制程序的流程。Eclipse 平台是一个开放的可扩展的集成开发环境。在该集成开发环境中可开发 EJB 组件、Java 程序、C 程序、Web 站点等各种不同类型的应用。

平台本身不会提供大量的最终用户功能。平台的价值在于它的促进作用：根据插件模型来快速开发集成功能部件。Eclipse 平台提供创建模块及建造和运行集成软件开发工具的基础。平台提供确定平台的基本功能的插件，通过附加插件进一步增强平台的其他功能。基本平台对任何对象而言总是一个集成开发环境。Eclipse 平台使工具开发商能独立开发工具并能和其他的工具无缝集成，用户在使用时甚至无法判断在何处一个工具结束，另一个工具开始。

2.1　平台基本机制

Eclipse 是集成开发工具的通用平台。Eclipse 具有一个基于微内核的、开放的和可扩展的体系机构。所有功能都通过插件（Plug-in）提供。

Eclipse 平台建立在插件机制上。插件是 Eclipse 平台功能的最小单元，可单独开发和提交。一个简单的工具通常作为单个插件编写，而一个复杂工具的功能则通过多个插件实现。除了称为平台运行时的小内核外，所有的 Eclipse 平台功能都由插件实现。Plug-in 机制将发现、集成和运行插件。我们将在“Eclipse 的插件技术”一章中详细讨论插件技术的原理、机制和实现。产品的各个插件一起组合成功能部件。

基于 Eclipse 的产品是作为一组插件构造的。每个插件都包含提供某些产品功能的代码。插件的代码和其他文件安装在本地计算机上，需要时自动激活。产品的各个插件一起组合成功能部件。功能部件的概念是 Eclipse 的新增内容；它类似 Eclipse 1.0 中的组件概念。本书在不引起混淆的情况下，将不严格区分功能部件和组件。

插件机制可用于划分 Eclipse 平台本身。实际上，不同的插件分别提供工作空间，工作台等。甚至平台运行时本身也有自己的插件。平台的非图形用户界面配置可能简单地忽略工作台插件及依赖它的其他插件。

Eclipse 平台的基本组件特性使得组件易于安装到基于 Eclipse 的产品，以及更新产品的现有功能部件和插件。可通过使用传统的本地安装程序从 Eclipse 独立运行或使用 Eclipse 平台本身的更新管理器实现此目的。Eclipse 更新管理器可用来从基于 Web 的特定的 Eclipse 更新站点中发现、下载和安装已更新的功能部件和插件。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Eclipse 平台的主要作用是向工具开发者提供一套机制和需要遵循的一组规则。工具开发者使用这套机制和规则可以创建无缝集成的工具。这些机制通过明确定义的 API 接口、类和方法来体现。平台提供一些有用的构件和框架，使用它们将有利于加快新工具的开发速度和提高新工具的开发效率。Eclipse 平台的设计和构造能满足如下需求：

- 支持用于应用开发的各种工具的构造。提供工具使常见的任务自动化。平台需要足够稳定，以便可以在它上面构建业界领先的工具。
- 对工具提供者没有任何限制，包括独立软件开发商（ISV）。以便用户可以选择最合适的工具。
- 支持处理任何类型的内容（如 HTML，Java，C，JSP，EJB，Word 文档、XML 和 GIF 等）的工具。
- 便于来自于不同工具提供者和处理不同类型内容的工具间的无缝集成。功能要足够强大，不需要额外的大量工作就可以支持集成。
- 应用开发环境既支持图形用户界面又支持非图形界面。
- 可运行于多种操作系统，如 Windows、Linux 和 Solaris 等。
- 利用广泛应用的 Java 程序语言来编写工具。并且足够简单，以便可以很容易地将现有工具迁移到 Eclipse 平台中。

Eclipse 平台定义了一个开放式体系结构，以便使每个插件开发小组专注于他们擅长的领域。让资源库专家构建后端，让可用性专家构建最终用户工具。Eclipse 的核心是动态发现插件的体系结构。平台负责处理基本环境的后台工作，并提供标准的用户导航模型。于是每个插件可以专注于执行特定的任务。添加重要的新功能部件和集成级别，不会影响其他工具。图 2.1 描述了 Eclipse 的平台结构。本章将分别介绍这些主要子系统和 API。

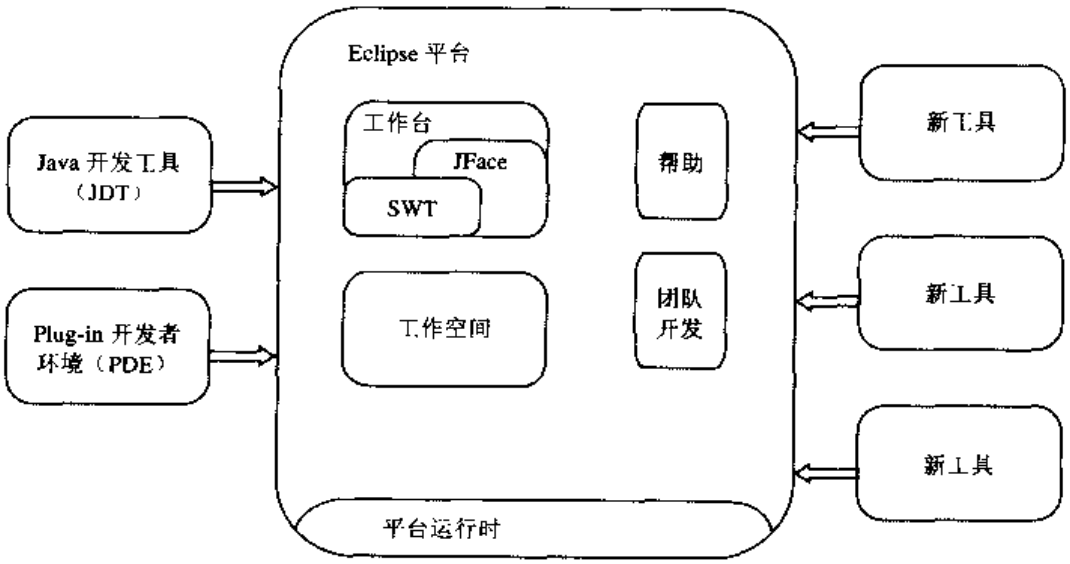


图 2.1　Eclipse 平台结构

平台中的每个主要子系统本身是由一组插件来构成的，这些插件实现一些重要功能并定义扩展点。扩展点是系统中严格定义的一些位置，插件可以在此添加功能。扩展点

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

也就是 Eclipse 为插件提供的接口。每一个插件都是在现有的扩展点上开发，当插件向扩展点添加实现时，我们就认为它向平台添加了扩展。插件可以定义它们自己的扩展点，以便让其他插件可以与它们紧密地集成在一起，或者只是将扩展添加至其他插件的扩展点。扩展机制是将功能添加到平台和其他插件的惟一方法，所有插件都使用相同的机制。Eclipse 平台是根据扩展点的这一概念来建立的。

Eclipse 软件开发工具箱中包含的两个主要工具：Java 开发工具和插件开发环境，对于插件开发很有用。我们将在后面的章节分别讨论。

在应用程序开发过程中，Eclipse 平台为应用开发人员和工具开发商提供了两个非常重要的特性：

- 开发人员可以使用任何插件工具。在同一环境中，开发人员可以有统一的用户接口、代码管理和测试体验。正是由于这个创新之处，开发人员在使用多个工具时，无须将处理结果在不同的工具间导入和导出，取而代之的是所有开发任务都可在同一环境中执行。Eclipse 平台使得不同的开发工具能够无缝集成，用户在使用时甚至无法判断在何处一个工具结束运行，另一个工具开始运行。
- 工具开发商可集中精力于所擅长领域的工具功能的开发。无须维护和支持使用工具所需的所有服务，如用户接口和对象存储。这个特征使得新的工具开发更加容易，并使得新开发的工具可以毫无困难地立即集成到已建立的平台中。

2.2 平台运行时

平台运行时（runtime）管理发现和启动所有已安装插件。每个插件只是在用户从操作或菜单项中选择它的功能时才被载入和初始化。

平台运行时定义扩展点和插件模型。底层平台运行时使用称为扩展点的扩展模型来允许插件开发者添加对其他文件类型和定制安装的支持。平台运行时核心实现运行时引擎，它启动平台基体，并动态地发现插件。插件使用清单文件来向系统描述它自己。

Eclipse 平台运行时提供动态扩充对象的机制。实现可适配接口的类声明它的实例开放给第三方扩展。一个可适配的实例可查询实现接口或类的适配器对象。通过 Eclipse 平台可以注册合适的适配器工厂（Factory）为可适配对象的已有类型添加操作（Behavior）。不同的开发者可根据各自不同的目的，分别独立扩展同一可适配对象。当请求一个给定接口的适配器时，平台识别并激发合适的适配器工厂来创建它。任一个插件可采用该机制为已有可适配对象增加操作，以及为其他插件的使用和扩展定义可适配对象的新类型。

启动工作台时，并未启动单个 Java 程序。在激活平台运行时的时候，它可以动态地发现已注册的插件并根据需要来启动它们。可以将插件安装和添加到注册表中但不激活插件，除非已经根据用户的活动请求了插件所提供的功能。平台负责维护已安装的注册表以及它们提供的功能，它动态地发现插件并维护平台注册表中的信息。由于运行时可以根据平台中的用户操作启动所需插件，因此不需为已安装但未使用的插件消耗内存或

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

影响性能。

2.3 工作空间（Workspace）

通过插件技术插入 Eclipse 平台的各种工具的操作范围是用户的工作空间，工作空间也称工作区。也就是说，工具在用户的工作空间的资源上操作。这些工具通过 API 处理工作空间的资源。工作空间（或称工作区）定义 API，以创建和管理由工具生成的并保存在文件系统中的资源，即项目、文件和文件夹。

工作区包含资源集合。项目是任意数目的文件和文件夹的集合，但是不包含其他项目。它是用来组织与特定区域相关的其他资源的容器。文件和文件夹就像文件系统中的文件和目录。文件夹包含其他文件夹或文件。文件包含任意顺序的字节，其内容不是由平台解释的。

工作区的资源组织成树型结构，项目位于顶部，而文件夹和文件在下面。特殊资源、工作区根目录资源充当资源树的根目录。工作区根目录是在创建工作区时在内部创建的，只要工作区存在，工作区根目录就存在。

工作区可以有任何数目的项目。工作空间由一个或多个顶层项目组成，每个项目映射到文件系统中相应的用户指定目录。虽然所有的项目都默认地映射到单个工作空间目录下，但在一个工作空间的不同项目可以映射到不同的文件系统目录或驱动器。底层操作系统中的标准程序和工具可以直接存取工作空间中的所有文件。因为使用可适配对象表示工作空间资源，这样其他开发者就可扩展这些资源的操作。

为了最大程度地降低偶然丢失文件的风险，一个底层的工作空间历史（history）机制跟踪先前由集成工具修改或删除的任何文件的内容。用户可通过基于空间和生命周期的选择设置来控制如何管理历史。

工作空间提供书签（marker）机制来注解资源。书签用于记录不同的注解如编译错误消息，待做的列表项书签，搜索次数和排错断点。书签机制是开放的。插件可以声明新的书签子类型并控制是否在运行期间保存它们。

平台提供一种通用机制，它使工具能够跟踪工作空间资源的更新。通过注册资源更新监听器，一个工具可确保接收到所有资源创建、删除及文件内容更新的事后通知。平台延迟事件通知，直到一批资源操纵操作的结束。事件报告采用资源增量（resource deltas）的树型结构来描述整个网络资源创建、删除和更新操作的结果。资源增量也提供有关书签更新的信息。

资源树增量对那些显示资源树的工具尤为有用和有效，因为每个增量指出工具可能在哪里需要增加、删除或刷新屏幕上的小部件。此外，因为许多半独立工具可以同时操作一个项目的资源，该机制允许一个工具检测邻近特殊文件或相关文件类型的活动。

工具如编译器和链接检查器必须对上千个不同的文件应用相同的分析和转换。平台提供了一个增量项目构建器框架；增量构建的输入是自上次建造以来捕获网络资源差别的资源树增量。复杂工具可以使用这个机制来提供可升级的解决方案。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

平台允许几个不同的增量项目构建器在同一项目上注册并提供触发项目和工作空间构建的途径。可选的工作空间自动构建功能使得在每个资源修改操作（或批量操作）之后自动触发必要的构建。

工作空间保存—恢复过程开放给参与的插件，以期在会话间保持与工作空间的同步。这两个阶段的保存过程确保各插件的重要状态作为原子操作写入磁盘。在后面的会话中，当单个的插件重新被激活并重新加入保存—恢复过程，将会产生一个工作空间资源增量，该增量描述了该插件所参与的自最后保存起的网络资源的差异。

2.4 工作台（Workbench）

术语“工作台”指的是桌面开发环境。工作台的目标是通过为创建、管理和导航“工作台”资源提供公共范例来获得无缝的工具集成。

从最终用户的观点看来，Eclipse 平台使用公共工作台模型来集成工具。工作台为工具开发者提供集成框架，在此框架上可构建其他更复杂的和专用的工具。工具开发者可以使用插件开发工具包创建他们自己的插件。可使用严格定义的扩展点将用户开发的工具插入到工作台中。

在工作台中，用户可浏览资源和使用工具插件，相应地工作台是浏览由插件提供的所有功能的地方。通过使用工作台，我们可以浏览资源并且可以查看和编辑这些资源的内容和特性。它定义用于添加图形用户界面的组件，例如视图或菜单操作的扩展点。

平台子系统通常将可视功能部件添加到平台中，并提供 API 来扩展它们的功能。这些组件中的某些组件提供与扩展点不直接相关，但可用来实现扩展的附加类库。工作台为用户提供全局结构和一个扩展的用户界面。工作台用户界面插件实现工作用户界面，并定义许多扩展点，这些扩展点允许其他插件向菜单和工具栏添加操作、拖放操作、添加对话框、添加向导，以及定制视图和编辑器。工作台用户界面插件还提供了对于用户界面开发很有用的框架。这些框架用来开发工作台本身。使用框架不仅容易开发插件的用户界面，而且可以确保插件具有常见的外观及一致的工作台集成。

工作台包含用于构建用户界面的附加工具箱 JFace 和 SWT。这两个图形用户界面组件设计为自包含的框架。工作台 API 及其实现是通过这两个工具箱构造的。

工作台 API 依赖于 SWT API，在较小的程度上依赖于 JFace API。Eclipse 工作台实现使用的是 SWT 和 JFace，而没有使用 Java AWT 和 Swing。

工作台为 Eclipse 平台提供专用的用户接口，并提供工具与用户交互的结构。因此工作台、Eclipse 平台，以及当平台运行时用户看到的主要窗口，在一般情况下含义相同。可以将工作台窗口看做是用来提供各种可视部件的框架。Eclipse 平台用户接口基于视图、编辑器和透视图，我们将在“Eclipse 工作台操作”一章中详细讨论。

从本质说，工作台并不是一个应用开发工具，而是一个开放工具平台。它主要是应用开发工具的运行平台和开发环境及集成技术使得工具提供商可以创建平台无关的、专

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

用的企业级应用开发工具，与工作台的其他工具无缝集成。

2.5 标准小窗口工具箱和图形用户界面框架

“标准小窗口工具箱”简称 SWT，它是 Java 开发者的小窗口工具箱。SWT 提供可移植的 API，并与底层本机 OS GUI 平台紧密集成，它是一个与本地窗口系统集成在一起的小部件集和图形库。JFace 是一个用户界面工具箱，它提供很难实现的、用于开发用户界面功能部件的 helper 类。JFace 使用 SWT 实现用户接口工具集，简化用户接口程序任务。SWT 和 JFace 都是 Eclipse 平台上的主要组件。

SWT 是一个低级别的、与操作系统无关的工具箱，它支持平台集成和可移植 API。SWT 集成于本机窗口系统，但有独立于操作系统的 API。许多低级别的用户界面（UI）编程任务是在 Eclipse 平台的较高层处理的。用户界面添加项的 plugin.xml 标记指定菜单和工具栏内容，而不需要进行任何 SWT 编程。JFace 查看器和操作提供了应用程序与小窗口之间的常见交互的实现。JFace 是用 SWT 实现的图形用户界面工具箱，它简化了常见的图形用户界面的编程任务。JFace 在其 API 和实现方面都独立于窗口系统。Eclipse 工作台、JFace 和 SWT 间的关系如图 2.2 所示。



图 2.2 Eclipse 工作台、JFace 和 SWT

SWT 和 JFace 之间的界线是很明显的。SWT 根本不依赖于任何 JFace 或平台代码。基于 SWT 可以构建独立的应用程序。JFace 用来在 SWT 库顶部提供常见的应用程序用户界面功能。JFace 并不试图“隐藏”SWT 或者替换它的功能。它提供一些类和接口，用于处理与动态用户界面编程相关联的许多常见任务。

2.5.1 SWT

因为 SWT 为小部件和图形提供了一个公用的独立于操作系统的 API，可以和本地窗口系统紧密地集成在一起，所以整个 Eclipse 平台用户接口及插入其中的工具，都使用 SWT 来向用户提供信息。

小窗口工具箱设计中的常见问题是可移植工具箱与平台集成之间的平衡，解决可移动工具集与本地窗口系统集成之间的不匹配。Java AWT（抽象窗口工具箱）提供较低级别的部件如列表、文本域和按钮，但没有像树或富文本（rich text）这样较高级别的部件。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

AWT 部件使用窗口系统上的本地部件来实现，这意味着基于AW 的编程只能使用所有平台上都可用的部件。

Java Swing 工具箱通过模拟对象树、表和 rich text 这样的部件，提供它们的非本机实现来解决这个问题。Swing 还提供外观和感觉模拟层，试图使应用程序与本地窗口系统相似。然而，模拟的部件在外观和感觉上永远滞后于本地部件，并且用户与模拟部件的交互往往很明显，这使得它很难超越专门为一特定本地窗口系统开发的应用程序。

SWT 通过定义在所有受支持的平台上提供的公共可移植 API 来解决可移植工具箱与平台集成之间的冲突，并尽可能使用本机部件（或称小窗口）在每个平台上实现该 API。公用的底层部件如列表、文本域和按钮在所有地方都是本地实现的。如果没有合适的部件，SWT 工具就提供适当的模拟。因此，工具箱能立即反映底层操作系统的图形用户界面的外观中的任何更改，同时在所有平台上维护一致的编程模型。

一些通常有用的高层部件在一些窗口系统上可能需要模拟。例如，SWT 工具条部件在 Windows 上以本地工具条部件实现，而在 Motif 上则作为模拟部件。这个策略使得 SWT 在所有环境中都能维持一致的编程模型，同时使得后台本地窗口系统的外观和感觉得到最大程度的表现。

对于所有平台都未提供而又未广泛使用的功能部件，可从 SWT 中省略。例如，Windows 提供了实现日历的小窗口，但是在 SWT 却没有提供。

SWT 还提供了仅在特定平台上的功能部件。特定于平台的功能部件分成单独的包，它在包中明确表示平台名以指出它是内置的、不可移的，以防其他平台误使用该功能部件，例如 Windows ActiveX。窗口系统特定 API 被隔离在特殊命名的包中。

SWT 与本地窗口系统的紧密结合不仅限于外观问题。SWT 还与本地桌面的一些功能，如拖放和删除等交互作用，与操作系统桌面应用程序集成，并能使用基于操作系统组件模型开发的组件，如 Windows ActiveX 控件。

SWT 为每个本地窗口系统分别提供不同的 Java 实现。Java 本地库是完全不同的，每个都带有后台窗口系统特定的 API。（与 Java AWT 相反，Java AWT 在一组 Java 本地方法公用集的 C 代码实现定位窗口系统的特定差别。）因为没有特殊的逻辑隐藏在本地，SWT 完全以 Java 代码实现。并且 Java 代码对于本地操作系统开发者来说更熟悉。Windows 程序员会发现能很快熟悉 Windows 的 SWT 的 Java 实现，因为它包含了他们在 C 编程时早就知道的 Windows API。一个 Motif 程序员看待 Motif 的 SWT 实现是同样的道理。这个策略极大地简化了实现、调试和维护 SWT，因为它允许所有相应的开发都用 Java 实现。当然，这没有直接涉及 SWT 的普通客户，因为这些完全隐藏在窗口系统独立的 SWT API 中。

2.5.2 JFace

JFace 是一个 UI 工具箱，它提供很难实现的、用于开发 UI 功能部件的 helper 类。JFace 在原始的小窗口系统的级别之上运行。JFace 用户界面框架提供较高级别的应用程序构造，用于支持对话框、向导、操作、用户首选项和小窗口管理。JFace 是处理许多

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

公用用户接口编程任务类的用户接口工具集。JFace 在它的 API 和实现上独立于窗口系统，并设计为与 SWT 共同工作。

JFace 提供用于处理常见的用户界面的编程任务的类。JFace 包含图像、字体注册表、对话、首选项和向导框架以及长时间运行操作的进程报告的常用用户界面工具箱组件。它的两个主要功能部件是操作和查看器。查看器负责处理填充、排序、过滤和更新小窗口等工作。操作和添加项介绍用于定义用户操作的语义，并指定在何处提供它们。图像和字体注册表提供用于处理用户界面资源的常见模式。对话框和向导定义用于与用户进行复杂交互的框架。

通过定义扩展点工作台插件可添加平台的用户界面的功能。许多这些扩展点，特别是向导扩展，是通过使用 org.eclipse.jface.*包中的类来实现的。JFace 结束和工作台开始的位置的界线有时候并不明显。通常，JFace API（来自于包 org.eclipse. jface.*）与工作台扩展点和 API 无关。可以想像，根本不必使用任何工作台代码就可以编写 JFace 程序。

工作台使用 JFace，但是又试图尽可能地减少相关性。例如，工作台部件模型被设计为独立于 JFace，例如可以直接使用 SWT 小窗口来实现视图和编辑器，而不必使用任何 JFace 类。工作台尽可能保持“JFace 中立”，允许程序员使用他们觉得有用的 JFace 的部件。

JFace 使得开发者可以专注于实现特定插件的功能，而不必花费精力来处理底层小窗口系统，无须直接具体解决那些应用程序中的常见问题。

2.6 团队协同开发

团队协同开发定义用于管理资源和设置资源版本的小组编程模型。团队协同开发有时也称为小组开发。Eclipse 的“小组”插件允许其他插件为小组编程、资源库访问和版本化定义和注册实现。Eclipse SDK 包括一个 CVS 插件，它使用小组支持在 SDK 中提供 CVS 客户机支持。

在 Eclipse 中小组插件定义附加的 API，允许插件集成版本和配置管理资源库的功能。资源库提供的功能将从根本上影响用户工作流程，因为有附加的步骤检索文件、与本地内容比较其内容、为它们进行版本控制，并将更新的文件返回至资源库。小组插件 API 允许资源库插件提供程序定义它们自己的工作流程，这样熟悉其产品的用户可以用相似的方式使用平台。

对于工作空间中的项目，Eclipse 工作台的“小组”支持插件提供了版本和配置管理能力，并向开发者提供了版本和配置管理所需的视图。大多数工具仅操作在工作空间的资源上。

Eclipse 的小组开发模型具有以下特征：

- 支持多个异构的资源库
- 支持轻量级小组开发模型

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 支持资源的版本化

资源库在共享服务器上存储需要进行版本管理的资源。Eclipse 平台可支持多种存储类型。单个工作空间可以同时存取不同类型的资源库。

在 Eclipse 中，小组成员共享的一个或多个相关项目的配置及他们的文件夹和文件称为流（stream）。开发者团队共享反映他们正在进行的工作及每日集成在一起的修改的流。实际上，流是在资源库中的一个共享工作空间。

每个小组成员在各自的工作空间工作，并在资源的私人拷贝上进行修改。小组其他成员不能马上看到这些修改。在方便的时候，开发者可以将他们的工作空间与流同步。当一个小组成员生成了新工作，将这些修改发布到流就可以将该工作与团队的其他成员共享。

类似地，当一个团队成员希望获取合适的最新工作，他们就获取由其他人向流发布的修改。两个同步操作可在资源子树上有选择地执行，并可预览检进和检出的修改。

一个资源库通常可有任意数目的流，由名字加以区分。例如，在 CVS 资源库中，一个流映射到主干（“HEAD”）或分支上。流可作为存储和发布开发者个人工作和在小组主开发流外的早期原型版本的轻量级容器。当同一项目出现在不同的流中，资源各自独立地演化；发布到一个流的修改对其他流没有影响。

资源库还支持不变资源版本的概念。例如，在 CVS 资源库中，一个项目版本对应分配给文件版本集一个独特的标签。当修改发布到流时，就自动创建了更新文件的新版本。新项目版本可以作为流或工作空间中项目的当前版本的快照。

Eclipse 支持异构资源库和存储类型。可在不同类型的资源库中管理工作空间中的每个项目。工作空间为项目资源记录流特定的同步信息。

小组开发的支持与工作台间无缝集成。在 Eclipse 工作台中包括支持小组开发的透视图，以及支持版本和配置管理的各类视图等。

2.7 帮助

联机帮助系统让用户能够浏览、搜索和打印系统文档。帮助系统还提供了用于按关键字查找所需信息的文本搜索引擎和用于查找描述正在使用的特定功能信息的上下文相关帮助。帮助系统的高级特性包括具有参考链接（称为弹出信息）的上下文相关帮助和从文档调用平台代码的能力。

当在 Eclipse 工作台的主菜单中选择“帮助”→“帮助内容”首次打开帮助浏览器时，显示的第一个视图称为“内容”。“内容”视图显示产品文档的各种分组。单击其中的任何一个链接都可以展开该组文档的导航树。浏览该树并单击主题则可以显示它们。

如果想要在联机帮助中查找特定的信息，可使用“搜索”功能。帮助系统将搜索整个帮助文档集或仅搜索其中的某个部分以查找符合指定条件的主题。

如果正在完成某个任务但遇到了用户不了解的界面部分，则可以使用弹出信息。通过单击所讨论的界面小窗口或者使用 Tab 键来使该小窗口处于焦点中，然后按 F1 键。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

弹出信息将提供有关界面的信息以及指向更多的相关信息的链接。

Eclipse 平台的帮助设施为开发者提供了原始构建块来构造和添加平台文档。它不会限制文档的结构或粒度。开发者可以为文档选择符合需要的工具和结构。帮助插件允许使用目录（toc）文件描述平台的文档结构。

“帮助”插件实现平台优化的帮助 Web 服务器以及文档集成设施。它定义了一些扩展点，插件可以使用这些扩展点添加帮助或者添加其他插件文档作为可浏览的书籍。文档 Web 服务器包含特殊的设施，它允许插件使用逻辑的，基于 URL 的插件而不是文件系统 URL 来引用文件。

使用 org.eclipse.help.toc 扩展点添加插件的联机帮助。可以作为代码插件的一部分来添加联机帮助，也可以在它自己的文档插件中单独提供联机帮助，因此可提供其他功能部件来集成产品级别文档配置中的帮助主题。

在代码和文档小组属于不同的组，或者开发者想要减少文档与代码之间耦合的情况下，将文档分成单独的插件将是一种很好的方式。

通过 Eclipse 平台帮助机制，可向一个或更多在线书（在线书指程序附带的文档和参考说明等）定义和提交文档。例如，一个工具通常向《用户指南》提供帮助类型文档，向《程序员指南》提交 API 文档（如果有的话）。原内容以 HTML 文件形式提交。将原内容以合适的浏览结构安排进在线书的方式，在 XML 文件中分别表示。这种分离使已存在的 HTML 文档不需编辑或重写就可直接合并进在线书。

附加的浏览结构将标题以树的形式来表示书的内容。每个标题，包括扉页标题，可以有一个原内容页和它连接。一本书可以有多个顶级标题的不同列表，使得一些或所有的相同信息以完全不同的组织形式表现，如按任务或工具来组织。

XML 浏览文件和 HTML 内容文件存储在插件的根目录或子目录中。小工具通常将它们的帮助文档作为代码放在同一插件中。大工具通常有分别的帮助插件。平台使用自己的内部文档服务器从文档 Web 提供真正的 Web 页面。定制服务器允许平台解决特殊插件间的连接，并从 Zip 压缩中解压 HTML 页。

在组织帮助系统时，只有在被归档的工具集被关闭后才可能生成完整的标题树。在 Eclipse 平台中，工具集是无限制的，并且帮助文档的结构需要标准化。平台帮助机制允许工具提交原内容和标题集，并指出如何在预定义的插入点将它的标题插入到已存在的标题树中。

2.8 外部工具

Eclipse 支持从内部运行任何外部工具，并将这些工具的输出显示在日志控制台视图中。Ant 也支持利用此新功能部件。外部工具允许开发者使用 Eclipse 工作台配置和运行程序、批处理文件和 Ant 构建文件等。还可以保存这些外部工具配置并在以后运行它们。可以将外部工具作为项目构建进程的一部分来添加。每当构建项目时，这些外部工具将以指定的次序运行。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com

本教程由站长百科收集整理

总地来说，Eclipse 平台提供了通用构造块内核和 API，如工作空间和工作台以及新功能可以被集成的各种扩展点。通过这些扩展点，开发者可用编写不同的插件工具扩展 Eclipse 平台，展现给用户的是插件集构成的集成开发环境。工具可以定义新的扩展点以及自己的 API，从而作为构造块和集成点为其他工具服务。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第3章 工作台

工作台指的是 Eclipse 桌面开发环境，每个工作台窗口都包含一个或多个透视图。透视图则包含视图和编辑器，并且控制出现在某些菜单栏和工具栏中的内容。工作台提供了使用 Eclipse 工具必需的用户界面。工作台的用户界面通常通过透视图、视图和编辑器等方式表示。

通过工作台，可以创建、管理和导航工作台资源，获得无缝工具集成。Eclipse 工作台的操作界面包含窗口、菜单栏、工具栏等。Eclipse 软件开发工具箱界面友好，用户可以通过菜单、按钮的操作，方便、快捷地实现各种功能。我们将存在于“工作台”中的项目、包、文件夹和文件统称为资源。

3.1 Eclipse 工作台概述

由于 Eclipse 平台中可自由地加入各种插件，因此在不同的应用中窗口外观往往有所不同。图 3.1 是 Eclipse 软件开发工具箱首次运行后的图形用户界面，仅包含 Eclipse 平台的一些标准的基本组件。

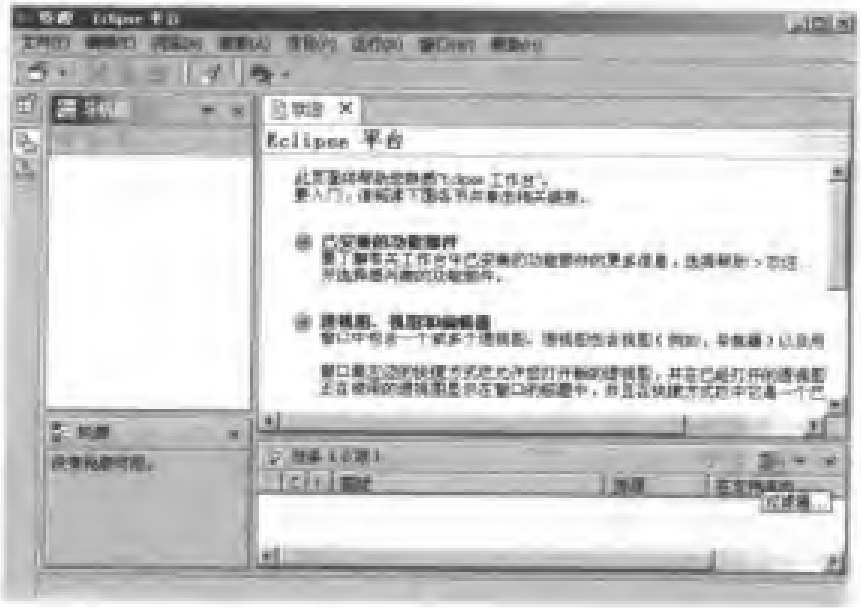


图 3.1 Eclipse 首次运行界面

工作台窗口显示一个或多个透视图。最初显示的是“资源”透视图。透视图包含一些视图，如导航器和编辑器等。可同时打开多个工作台窗口。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在图 3.1 中，左上为导航器（Navigator）视图。导航器视图中显示了用户工作空间（Workspace）中的所有文件，它显示了关于工作台的内容以及层次结构中的资源如何全部互相关联的信息。导航器视图提供了资源的分层视图。用户使用导航器可以打开资源并进行编辑。单击导航器视图中的文件，右上的正文编辑器则显示相应文件的内容。

窗口的最左边是一个快捷方式栏，它允许您打开新的透视图并在已打开的各透视图之间移动。活动透视图的名称显示在窗口的标题中，并且在快捷方式栏中它的图标是已按下的图标。

窗口左下为大纲（Outline）视图，或称为轮廓视图。大纲视图显示了正在被编辑的文本视图的概要，但如果正在被编辑的是文本文件，则文本视图中不显示该文本文件的概要。右下是任务栏（Tasks）视图，显示任务列表。

3.1.1 透视图

完成一个应用程序需要许多任务。具有不同技能和任务的人员共同完成应用程序的设计、开发、测试、调试和部署。Eclipse 平台为不同的开发任务提供了专用的用户界面透视图。

透视图是加到环境中的一组工具视图和数据视图的集合，将这些视图组织成透视图，将更容易执行特定任务。换言之，透视图可以看做为工作台窗口中的一组视图和编辑器。例如，一个 Java 透视图的用户界面可能包括 Java 包视图，Java 源代码编辑器和显示所有重大错误的视图。透视图是预先定义的，然而用户对透视图可以进行配置。

透视图控制出现在特定菜单和工具栏中的内容。它们定义可视操作集，开发者可以更改这些操作集来定制透视图。开发者也可以保存以此方式构建的透视图，并建立开发者自己的定制透视图，在以后可以打开它们。

图 3.1 显示了工作台（Workbench）窗口。工作台窗口显示一个或多个透视图（Perspectives）。透视图定义工作台窗口中视图的初始设置和布局。在窗口中，每个透视图共享同一组编辑器。每个透视图提供一组功能，以完成特定类型的任务或使用特定类型的资源。例如，Java 透视图将编辑 Java 源文件时常用的视图组合在一起，而“调试”透视图包含将在调试 Java 程序时使用的视图。透视图中包含多个视图和编辑器，如图 3.1 所示的导航器、正文编辑器、任务列表等。默认的 Java 透视图由四个视图组成。

① 包浏览视图和层次视图位于左边。

② Java 编辑器视图在中间。

③ 大纲视图位于右边。

④ 任务视图占据了窗口下面的大部分。

一个工作台窗口可以有几个不同的透视图。最初打开的是资源透视图，可以随时打开多个不同的透视图，在任一给定时刻只有一个是可视的。在一个透视图中同时可以打开几个不同类型的视图和编辑器。每个透视图有自己排列好的视图和编辑器，平铺、堆叠或分开在屏幕上显示。

如图 3.1 所示，在工作台窗口的最左边是快捷栏（Shortcut Bar），使用快捷栏可以打

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

开多个透视图。单击不同的透视图按钮，用户可以在这些透视图之间进行切换，激活所需的透视图，即用户可以在执行某个任务时切换到最适合该任务的透视图。所激活的透视图的名称将显示在工作台窗口最上面的标题栏中。同时，位于窗口最左边的相应的透视图快捷栏图标呈现图 3.2 所示的“按下”状态，其他所有透视图快捷栏则呈“浮起”状态。



图 3.2　资源透视图快捷栏的不同状态

在一个工作台窗口中，不同的透视图所包含的视图可能有所不同，但所有的透视图共享所有的编辑器。

可以利用不同的方法打开工作台窗口内的新透视图：

- 使用快捷方式栏上的“打开透视图”按钮。
- 从“窗口→打开透视图”菜单中选择透视图。

如图 3.3 所示，可以选择所需打开的透视图，如果要找的透视图在列表上看不到，则选择“其他”，这将弹出一个列出所有可用的透视图的“选择透视图”对话框。“选择透视图”对话框显示所有预定义的以及任何已经自定义（也称定制）并保存的透视图。选择希望加到工作台的那个透视图，然后单击“确定”。



图 3.3　打开透视图

Eclipse 将会记住最后一次关闭工作台时哪些透视图是打开的。在下次启动 Eclipse 时将再次打开这些透视图，在最前面显示的是上一次最后一次显示的透视图。

默认透视图总是显示在“窗口→打开透视图”菜单上。资源透视图是预定义的默认透视图。如要更改默认透视图，步骤如下。

① 从主菜单栏中，选择“窗口→首选项”。

② 展开左边的工作台类别，并选择透视图。打开透视图首选项页面。

③ 选择想要定义为默认值的透视图，并单击该透视图成为默认值。默认值指示符便

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

移至所选择的透视图。

④ 单击“确定”按钮。

平台为通用资源浏览、在线帮助和团队支持任务提供标准透视图，其他透视图由其他插件提供。在 Eclipse 中，其他的一些透视图包括：

- CVS 资源库浏览
- Java
- Java 浏览
- 安装 / 更新
- 插件开发
- 调试
- 资源

许多透视图共享相同的视图，因此同一功能无须在透视图间跳转。另外，还可以定制一个已有的透视图以满足任务需求或仅仅是为了满足用户喜欢的工作方式。这些新建透视图可以按它们自己的名字保存并在需要时使用。编辑器视图由所有的透视图共享，因此可以从每个打开的透视图中访问任何一个打开的文件。

3.1.2 视图

视图是工作台中一个可视化组件。视图提供了用户正在工作台中使用的一些对象的信息。通常用它来浏览分层信息如工作台中的资源、打开编辑器或显示活动编辑器的属性。当用户选择工作台中的不同对象时，视图通常就会更改它们的内容。

在一个工作台窗口中仅可以存在某一特定类型视图的一个实例（Instance）。视图可以是活动或者不活动的，但只能有一个视图处于活动状态，其他的视图处于非活动状态，处于活动状态的视图的标题栏将突出显示（Highlight）——蓝色显示，而处于非活动状态的视图的标题栏将成白色。如单击某一视图，则该视图将处于活动状态。

视图可以具有它们自己的菜单。如要打开视图的菜单，可单击视图标题栏左端的图标。某些视图还具有它们自己的工具栏。视图工具栏上按钮表示的操作仅对该视图中的各项起作用。

视图可能会单独出现，也可能与其他视图一起以选项卡的形式进行叠放。工作台提供了几个标准视图。在工作台窗口中，可通过打开和关闭视图以及通过将它们放在不同的位置来更改透视图的布局。

可以通过拖拉操作重新安排透视图上的视图。还可以通过双击标题栏将视图最大化，再次双击标题栏使它恢复到原始大小和位置。可以将透视图重新安排到不再使用的地方。通过在主菜单条上选择“窗口（Window）”→“重新设置透视图（Reset Perspective）”，透视图可以很容易地恢复到它们的原始状态。

视图通常用来导航层次信息、打开编辑器或显示处于活动状态的编辑器的特性。视图中的任何修改都会立即被保存。可以使用“窗口→显示视图”菜单来显示所要显示的视图。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

导航器视图显示了工作台的内容以及层次结构中的资源如何全部互相关联的信息。从这里，可以打开文件进行编辑，或选择资源进行某些操作。右键单击导航器视图中的任何资源可以打开弹出菜单，该菜单允许用户执行诸如复制、移动、创建新资源、将资源相互比较等操作。如要查看每个菜单项执行的操作的描述，使选择突出显示移至该菜单项并按 F1 键。默认情况下，导航器视图包括在资源透视图中。

任务视图显示工作台中的所有任务和问题，包括与特定文件（甚至特定文件中的特定行）相关联的任务和未与任何特定文件相关联的一般任务。当在工作台中使用资源时，各种构建器可将问题、错误或警告自动记录在任务视图中。问题标记表示无效的状态（即错误、警告信息）。各种问题都被看做是任务并可显示在“任务”视图中。错误标记通常用来指示语法或编译错误的源位置。错误被看做是问题任务并可显示在“任务”视图中。例如，当保存包含语法错误的 Java 源文件时，这些错误就会记录在“任务”视图中。

当双击表示问题、错误或警告的图标时，关联资源的编辑器自动打开至相应的代码行。还可以手工将各条目添加至任务视图，例如，记录提示以便在以后某些事项中遵循执行。手工添加任务时，可选择将该任务与资源相关联，以便可以使用“任务”视图来快速地打开该资源进行编辑。可以在任务视图中进行排序和过滤，例如，只查看优先级高的任务或只查看与特定资源或资源组相关联的警告和错误。默认情况下，任务视图包含在资源透视图中。要将“任务”视图添加至当前透视图中，单击“窗口→显示视图→任务”。

除了导航器视图和任务视图，Eclipse 视图包含大量其他类型的视图，使用较多的有书签视图、大纲视图、属性视图等。书签是导航经常使用资源的简便方法，书签视图显示工作台中的所有书签。大纲视图显示当前在编辑器区域中打开的结构化文件的轮廓，并列示结构化元素。属性视图显示所选项（例如资源）的属性名和值。

大纲视图显示当前在编辑器区域中打开的结构化文件的大纲并列示结构元素。结构化元素有类、字段和方法。大纲视图的内容是特定于编辑器的。工具栏的内容也是特定于编辑器的。

属性视图显示所选项，如资源等的属性名和值。开发者可以使用工具栏按钮切换为按类别显示属性或过滤高级属性。开发者还可以使用工具栏按钮将所选属性恢复为其默认值。

3.1.3 编辑器

编辑器是工作台中的另一个可视化组件。编辑器通常用来查看或编辑资源，编辑器允许用户在工作台中编辑某些内容。在一个窗口中，每个透视图可具有一组不同的视图，但所有透视图共享同一组编辑器。

编辑器是以文档为中心的，它很像文件系统编辑器。与文件系统编辑器相似，它们遵循“打开—保存—关闭”这一循环。与文件系统编辑器不同的是，它们与工作台紧密集成。平台为文本资源提供标准编辑器；更多的特殊编辑器由其他插件提供。

可以将不同的编辑器与不同类型的文件相关联。例如，当双击要编辑的文件来打开

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

它时，就会在 Eclipse 工作台中打开相关联的编辑器。编辑器域将显示和所编辑的文件对应的编辑器。如正在编辑一个.TXT 文本文件，则编辑器域则显示一个正文编辑器，所编辑的文件的文件名将显示在编辑器的选项卡中，如图 3.4 所示。



图 3.4 编辑文本文件

所有文件的显示和编辑都包含在编辑器窗口中。默认情况下，打开的多个文件以标签方式在同一个窗口中排列，可以用拖动方式将这些文件排列成各种布局。方法是拖动某一文件的标签到编辑器窗口的边框，当光标有相应变化时再释放。可以将编辑器叠放在编辑器区域中并通过单击编辑器的选项卡来激活每个编辑器。还可以将编辑器并排平铺在编辑器区域中，以便可以同时查看它们的内容。

可以有两种不同的方式打开编辑器。如使用该文件类型的默认编辑器查看或编辑，直接在导航器中双击该文件，或者选择该文件并从它的弹出菜单中选择“打开”。在 Eclipse 中，通过工作台首选项页面的有关设置，可以选择单击而不是双击来激活文件。如果想要使用特定编辑器来查看或编辑，则应首先在导航器中选择该文件，并从弹出菜单中选择“打开工具”选项。Eclipse 工作台将记住用户用来编辑特定文件的上一个编辑器，这使得较为容易地一直使用同一个编辑器编辑一类资源。

如标记中显示的文件名最左边出现*，则表明该文件编辑后没有保存。假如此时关闭编辑器或退出工作台，将会提醒用户保存所作的改动。有两种方式可以关闭编辑器：

- 单击任何打开的编辑器选项卡中的关闭按钮（“×”）
- 从菜单栏中选择“文件”→“关闭”

此外“文件”菜单栏还有一个“全部关闭”选项。这是关闭所有打开的编辑器的快速方法。如果选择“文件”→“全部关闭”，则将会提示用户选择想要保存哪些带有未保存更改的编辑器。

与视图类似，可同时打开任意数目的编辑器，但只能有一个编辑器处于活动状态，其他的编辑器处于非活动状态。处于活动状态的编辑器的标题栏将突出显示。处于活动状态的编辑器将可进行类似插入、删除、拷贝和粘贴等常规操作。

与视图不同，工作台窗口中可以存在编辑器的多个实例。用户可以以层叠方式显示

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

多个编辑器。默认情况下，编辑器叠放在编辑器区域中，但可以选择平铺它们以便同时查看源文件。通过单击不同编辑器的标记激活相应的编辑器。用户也可以在编辑器于中一起显示所有的编辑器。

有时候，能够最大化编辑器是很有用的。最大化视图和编辑器都是很容易的。要将编辑器最大化，双击该编辑器选项卡或从可从编辑器选项卡获取的弹出菜单中选择“最大化”。将编辑器恢复为其原始大小是以类似方式完成的，即双击编辑器标题栏或从菜单中选择“恢复”。

有时，可能需要使用外部程序来编辑工作台中的文件。例如，当工作台没有用于该文件类型的编辑器时。如果外部程序注册为该文件类型的默认编辑器，并且在“工作台”中没有为该文件类型注册其他编辑器，则会使用该外部程序。如果所打开的资源没有关联的编辑器，则工作台将试图发现一个合适的外部编辑器。对于这些和 Eclipse 平台没有紧密集成的外部编辑器，在工作台的编辑器域将不会显示。但在 Windows 系统中，支持 OLE 文档方式的编辑器也可以在工作台的编辑器区域中运行。

在 Eclipse 中可以指定工作台中文件类型所对应的编辑器，即将编辑器与工作台中的各种文件类型相关联。

① 从主菜单中，选择“窗口”→“首选项”。

② 在“首选项”窗口中，展开“工作台”类别，并选择“文件关联”，如图 3.5 所示。



图 3.5　在首选项中设置文件关联

③ 从文件类型列表中选择文件类型，或者单击“添加”按钮添加尚未存在于列表中的类型。

④ 在相关联的编辑器列表中，选择想要与该文件类型相关联的编辑器。

⑤ 单击“添加”按钮，打开“编辑器选择”对话框，如图 3.6 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 3.6 编辑器选择

⑥ 选择内部编辑器或外部程序，这取决于编辑器是为“工作台”构建的还是在“工作台”外部运行。

⑦ 如果选择外部程序，则可以单击“浏览”按钮来浏览文件系统。

⑧ 从列表中选择编辑器，并单击“确定”按钮。

⑨ 单击“确定”按钮完成将编辑器与所选择的文件类型相关联。

如图 3.6 所示，编辑器的类型包括内部编辑器和外部程序。当打开资源时，工作台会先查阅该资源的已注册编辑器的列表。如果找不到该资源的注册编辑器，工作台会检查底层操作系统，以确定是否为特定文件类型注册了编辑器。如果已定位编辑器，“工作台”将自动启动该编辑器。此类型的编辑器称为外部编辑器，因为它未显示为 Eclipse 工作台中的编辑器选项卡。这些外部编辑器并未与工作台紧密集成，也没有嵌入到工作台的编辑器区域中。如果未向底层操作系统注册文件的编辑器，Eclipse 工作台将尝试使用它自己的默认文本编辑器。

在 Windows 系统中，工作台支持 OLE 文档编辑器。工作台将尝试在适当位置启动编辑器以作为 OLE 文档编辑器。例如，如果机器上已经安装了 Microsoft Word，并且在操作系统中已将 Microsoft Word 注册为 .doc 文件的编辑器，则编辑 DOC 文件将导致在工作台内的适当位置打开 Microsoft Word；如果尚未安装 Microsoft Word，则将打开写字板。

3.2 Eclipse 平台的菜单

当视图处于活动状态时，工作台中与视图相关的菜单条将生效，而其他菜单条，如和编辑器相关的菜单条将处于失效状态。类似地，当编辑器处于活动状态时，工作台中与编辑器相关的菜单条将生效，而和视图相关的菜单条将处于失效状态。

视图有两类菜单。第一类位于视图的左上角，如图 3.7 所示。在不同的视图中，这些菜单项是相同的，用户用同样的方式使用这类菜单操纵视图。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第二类菜单如图 3.8 所示。用户单击下拉按钮选择菜单项。

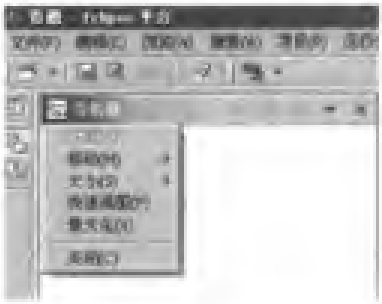


图 3.7　视图的第一类菜单

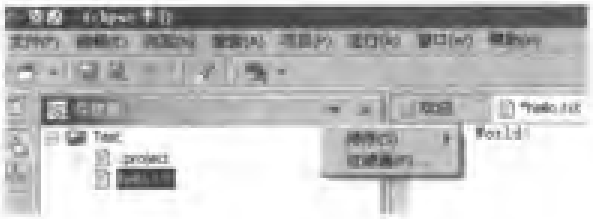


图 3.8　视图的第二类菜单

如图 3.9 所示，用户可以使用菜单操作“窗口”→“显示视图”选择视图。



图 3.9　使用菜单操作选择视图

3.3　Eclipse 平台的工具栏

工作台中有三种工具类型：主工具栏、视图工具栏和快捷方式栏。

主工具栏，有时称为工作台工具栏，显示在工作台窗口顶部，就在菜单栏下面。此工具栏的内容将视活动透视图的不同而有所不同。可根据活动视图或编辑器的状态启用或禁用工具栏中的项。在 Eclipse 中，可使用鼠标重新排列主工具栏的各个部分。

视图工具栏出现在视图的标题栏中。视图工具栏中的操作只适用于这些操作出现在其中的视图。某些视图工具栏包括一个包含该视图操作的菜单按钮（显示为一个倒三角形）。

快捷方式栏是沿工作台窗口左边缘出现的垂直工具栏。快捷方式栏允许快速访问当前打开的透视图。它还包含快速视图，可将这些视图创建为最常使用视图的快捷方式。

可出现在工作台工具栏、视图工具栏和快捷方式栏中工具按钮的含义如表 3.1 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

表 3.1 出现在工作台工具栏、视图工具栏和快捷方式栏中工具按钮的含义

按 钮	描 述	按 钮	描 述
	打开新的透视图		保存活动编辑器内容
	保存所有编辑器的内容		用新的名称或位置保存编辑器内容
	打开搜索对话框		打印编辑器内容
	打开资源创建向导		打开文件创建向导
	打开文件夹创建向导		打开项目创建向导
	打开导入向导		打开导出向导
	运行增量构建		运行程序
	调试程序		删除选择的项或内容
	将选择剪切到剪贴板中		将选择复制到剪贴板中
	从剪贴板粘贴选择		撤销最近的编辑
	重做最近撤销的编辑		导航至列表中的下一项
	导航至列表中的上一项		向前浏览
	向后浏览		向上一级浏览
	浏览回起始位置		打开视图的下拉菜单
	关闭视图或编辑器		固定编辑器以防止自动重用
	固定视图（使它不再是快速视图）		添加书签或任务
	过滤任务或特性		转至编辑器中的任务、问题或书签
	恢复默认特性		将项显示为树
	刷新视图内容		按字母次序对列表进行排序
	取消长期运行的操作		

3.4 创建项目

开发应用，首先须创建项目。项目是工作台使用的最大结构化单元。在 Eclipse 工作台中，所有资源都驻留在项目中。项目包含文件夹和资源，文件夹可包含其他文件夹和资源。

工作台提供了许多用于使用项目、文件夹和资源的机制。项目用于构建、版本管理、共享和资源组织。与文件夹相似，项目映射至文件系统中的目录。当创建某个项目时，将在文件系统中为它指定位置。

在工作台中，所有资源都驻留在项目中。项目中可以包含文件夹和文件。在 Eclipse 中创建 Java 项目是非常简单的操作。和 Eclipse 中的大多数任务一样，可有多种方式创建编程元素，可以用以下三种方式创建一个项目：

- 使用文件菜单
- 使用上下文导航菜单

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 使用向导工具按钮

3.4.1 使用文件菜单

通过文件菜单创建一个简单的项目，其过程如下。

① 选择“文件”→“新建”→“项目”，出现图 3.10。

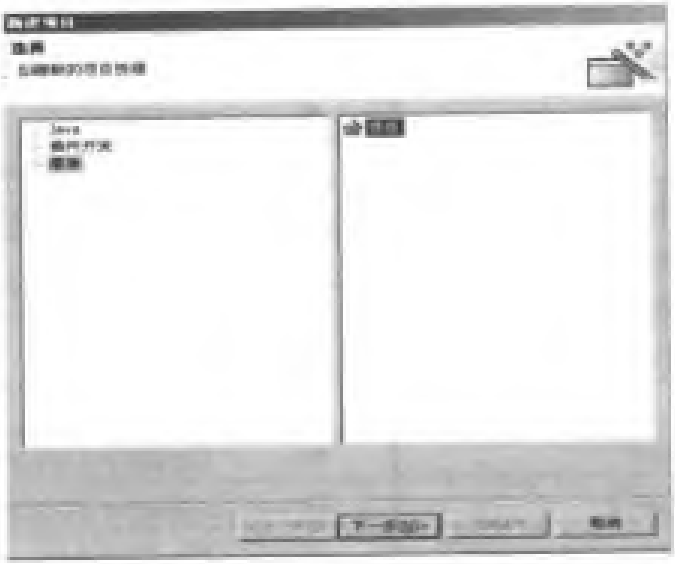


图 3.10 使用文件（File）菜单创建新的项目

② 在左边窗格中可选择相应的资源类型，假设选择“Java”，则在右边窗格中将显示“项目”，可选择它，然后单击“下一步”按钮，如图 3.11 所示。



图 3.11 指定项目名和路径

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在项目名域指定项目名，如例中项目名 Test，并进一步指定项目的保存路径，默认值为 Eclipse 平台安装路径下文件夹 workspce 中与项目同名的文件夹。大多数情况下无须修改默认目录，但如果需要也可以修改。

此时可以单击“完成”或“下一步”按钮。如单击“下一步”按钮则生成另一个对话框，如图 3.12 所示，在这里可以添加工程所需的库。可使用向导，将所需的 JRE 库 rt.jar 加到工程中。在这个对话框中可以定义其他的附加库。

为新的 Java 项目定义了相关值之后，单击“完成”按钮则完成新 Java 项目的创建。可以打开 Java 透视图查看向导创建的结果，导航器视图中将显示新建的项目，如图 3.13 所示。



图 3.12　Java 项目构建设置



图 3.13　项目创建完成后导航器视图

3.4.2　使用上下文导航菜单

如图 3.14 所示，通过弹出式菜单，也可以创建项目。



图 3.14　使用上下文弹出式菜单创建项目

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

3.4.3 使用新建向导工具按钮

如图 3.15 所示，我们也可以使用新建向导工具按钮创建项目。不管采用哪种方法创建新的 Java 项目，下一个对话框都如图 3.11 所示。

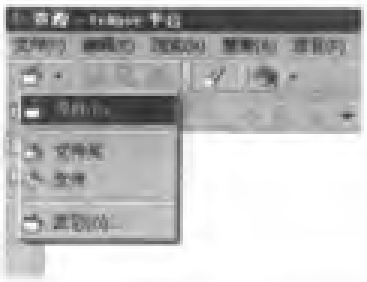


图 3.15 使用新建向导工具按钮创建项目

类似于创建项目，我们也可以使用菜单、上下文菜单和新建向导工具创建其他资源，如文件、文件夹等。一旦完成创建，这些资源将出现在导航器视图。

项目的状态为打开或关闭。当项目处于打开状态时，可以更改该项目的结构，且开发者可看到有关内容。Eclipse 平台在目录中存储开发者的项目内容。默认情况下，该目录在工作空间目录中并以项目名命名。

关闭项目后，就不能在工作台中更改它。被关闭项目的资源将不会出现在工作台中，但这些资源仍会驻留在本地文件系统上。关闭的项目需要较少的内存。由于在构建期间不会检查这些项目，所以关闭项目可以缩短构建时间。

当用户删除一个项目时，Eclipse 将询问用户是否要删除目录中的内容，即是否想删除它包含的所有文件和文件夹。默认值是不删除，但在许多情况下，需要整个删除内容和项目。但当开发者将项目不放在默认路径中时，可以保留相关文件，仅从 Eclipse 工作空间删除项目。删除文件类似于在文件系统中删除文件。

3.5 资源管理

安装 Eclipse 之后，在安装目录下会创建一个 workspace 文件夹。每当在 Eclipse 中新生成一个项目，默认情况下都会在 workspace 中产生和项目同名的文件夹，该文件夹将存放该项目所用到的全部文件。开发者可以使用 Windows 资源管理器直接访问或维护这些文件。

3.5.1 资源管理概述

资源是存在于工作台中的项目、文件夹和文件的统称。导航器视图提供了资源的分层视图并允许开发者打开它们进行编辑。其他工具可按不同的方式显示和处理这些资源。工作台中的资源有三种基本类型。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

（1）文件

与文件系统中的文件类似。

（2）文件夹

与文件系统中的目录类似。在 Eclipse 工作台中，文件夹包含在项目或其他文件夹中。文件夹可包含文件和其他子文件夹。

（3）项目

项目包含文件夹和文件。项目用于构建、版本管理、共享和资源组织。与文件夹相似，项目映射至文件系统中的目录。当创建某个项目时，将在文件系统中为项目指定位置。

在 Eclipse 工作台中按层次结构进行存储和显示。工作台中的资源内容的最上面一层称为根。包含其他资源的资源称为父资源，而包含在其他资源中的资源称为子资源。

项目中的文件夹和文件可以保存在文件系统中项目位置之外的位置，这些链接至文件系统中项目位置之外的位置的文件夹和文件称为链接的资源。使用链接的资源可以覆盖工作区中的其他资源，从而一个项目中的资源可以出现在另一个项目中。

为了限制视图中显示的资源集，或对一组元素执行操作，可对元素进行分组，称为工作集。例如，在导航器中选择了工作集，则只显示包含在该工作集中的资源。不同的视图提供不同方法来指定工作集。但是，它们通常使用“工作集选择”对话框来管理现有工作集以及创建新的工作集。

3.5.2 浏览和查找资源

资源是按层次结构存储和显示在 Eclipse 工作台中的。导航器视图显示了关于 Eclipse 工作台的内容和层次结构中各资源如何相互关联的信息。使用导航器视图中的“转至”操作很容易跳至特定资源，具体步骤如下。

① 选择“导航器”视图。

② 在菜单中选择“浏览”→“转至”→“资源”，将打开“转至资源”对话框，如图 3.16 所示。



图 3.16 选择资源

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

③ 在“转至资源”对话框中，在该对话框顶部的模式字段中输入资源名。资源名可以用通配形式表示，其中？代表任一字符，*代表 0、1 或任意多个字符。

④ 在输入资源名后，对话框会根据用户到目前为止已经输入的内容来过滤可能存在的匹配集合。

⑤ 从匹配资源字段中选择相应的资源，并单击“确定”按钮，导航器将选择相应的资源。

默认情况下，导航器视图将显示 Eclipse 工作台中的所有资源。可以选择项目或文件夹，并在上下文弹出菜单中选择“进入”，这将在导航器视图中隐藏所有其他资源，仅显示所选择的资源。

可以使用导航器中的工具按钮“后退”、“前进”和“向上”按钮，在显示所有资源和只显示所选资源之间进行切换，如图 3.17 所示。单击“上一步”按钮以显示所有资源。

在 Eclipse 工作台中，可以使用“搜索”按钮在导航器中的各资源中搜索文本字符串和文件。“搜索”按钮的图标如图 3.18 所示。

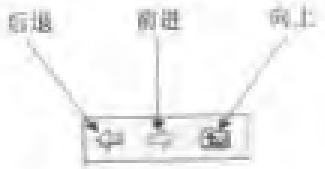


图 3.17　导航器中的工具按钮



图 3.18　“搜索”按钮

在 Eclipse 工作台按钮中搜索文本字符串的步骤如下。

① 在 Eclipse 工作台工具栏中，单击“搜索”按钮将打开搜索对话框，如图 3.19 所示。

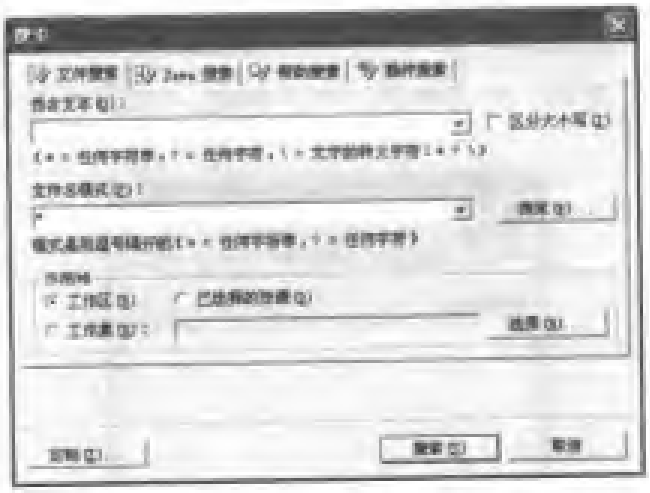


图 3.19　搜索窗口

② 在“包含文本”字段中，输入所需搜索的文本表达式。搜索表达式的可用通配符包括：

- “*”——匹配任何一组字符，包括空字符串。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- “?”——匹配任何单个字符。
- “\”——是文字的转义：如果想要搜索星号、问号或反斜杠字符，则在这些字符之前输入反斜杠以指示这些字符不是用做通配符，如“*”、“\?”或“\\”。

“包含文本”字段的组合框还可以让用户从最近执行的搜索的列表中进行选择。

③ 根据是希望执行区分大小写的搜索还是不区分大小写的搜索来选择或取消选择“区分大小写”复选框。

④ 在“文件名模式”字段中，可以指定要包括在搜索中的文件种类。单击“浏览…”按钮将打开“选择类型”对话框，如图 3.20 所示。



图 3.20 选择类型

文件名模式的可用通配符包括：

“*”——匹配任何一组字符，包括空字符串。

“?”——匹配任何单个字符。

⑤ “选择类型”对话框提供了一个从已注册扩展名列表中进行选择的快速方法。设置该对话框可以指定所要搜索资源的类型，如 txt 文本、Java 文件等，然后单击“确定”按钮。

⑥ 在“搜索”窗口的“作用域”字段中，可以指定要包括在搜索中的文件和文件夹。选项是：整个工作区、工作台中当前选择的资源，或作为已命名已定制文件和文件夹组的工作集。默认的作用域设置为工作区。

⑦ “搜索”窗口中“定制”按钮允许用户选择哪些搜索类型在对话框中可用。可以不更改此设置。

⑧ 单击“搜索”按钮将开始搜索，“搜索”对话框会显示搜索的进度。可以在搜索过程中单击“取消”按钮来取消搜索。

搜索结果将显示在搜索视图中，如图 3.21 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 3.21　搜索视图

在图 3.21 所示的搜索视图中，可以看到“HelloWorld”一词在工作区中出现 29 次（显示在视图的标题栏中），进一步看到，所有匹配的文件，以及每个文件中的匹配项的数目（默认值是一项匹配）。

搜索视图附带有多个工具按钮，如图 3.22 所示。



图 3.22　搜索视图上的工具按钮

单击“显示下一个匹配”按钮将导航至搜索表达式的第一个匹配项，且在编辑器区域中自动选择并打开相应的文件。再单击“显示下一个匹配”按钮，将依次导航到下一个匹配项，且自动选择并打开相应的文件。

从搜索结果中除去不感兴趣的匹配有时是很有用的。搜索视图的弹出菜单提供了两种方法来除去匹配（工具条中包含“除去所选匹配”按钮，但没有“除去当前匹配”按钮）：

① 除去当前匹配——除去用户当前正在查看的实际匹配。如果文件中包含多个匹配项，则只会除去当前匹配。

② 除去所选匹配——除去文件条目及其中的所有匹配。

单击工具栏中的“搜索”按钮可执行另一搜索。“搜索”视图将更新为显示新搜索的结果。可以使用“搜索”视图的工具栏上的下拉按钮来在两个搜索结果之间前后移动。选择再次搜索来重复初始搜索。

3.5.3　创建文件和文件夹

在 Eclipse 中，文件和文件夹是最主要的资源。在 Eclipse 中，创建文件的步骤如下。

① 在导航器视图中，右键单击想要在其中创建新文件的项目或文件夹。

② 从弹出菜单中，选择“新建”→“文件”。

③ 指定文件的名称，包括文件扩展名。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

④ 单击“完成”按钮。

创建的文件将在与其类型相关联的编辑器中打开。在 Eclipse 工作台中创建文件夹的步骤与创建文件类似，差别只是在弹出菜单中选择“文件夹”。

3.5.4 导入导出资源

为了共享资源，经常需要将操作系统中文件系统的文件导入到工作台或将工作台中的文件导出到文件系统。

1. 导入文件

Eclipse 提供导入向导，可以将外部文件导入到工作台。具体步骤如下：

① 选择需要导入文件的项目。

② 在主菜单中选择“文件”→“导入”，或在所选项目的上下文弹出菜单中选择“导入”。

③ 在导入向导中，选择“文件系统”，然后单击“下一步”按钮。

④ 如图 3.23 所示：



图 3.23 从本地文件系统导入资源

- 在目录字段中，输入或进行浏览以选择包含导入文件的目录。
- 在左窗格中选择文件夹会将文件夹的整个内容导入到工作台中，可以使用“过滤器类型…”按钮来过滤想要导入的文件的类型。在右窗格中，可选择一个或多个需要导入的文件。
- 选择导入资源目标的文件夹字段为项目的名称，用户也可以单击“浏览…”按钮来更改目标项目。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 在“选项”区中，可以选择以下选项之一：
 - 覆盖现有的资源而不发出警告
 确定导入资源时是否应不进行提示就覆盖已存在于工作台中的资源。如果此选项处于关闭状态，则在覆盖给定资源之前将会有提示，在这种情况下，开发者可以覆盖该资源、跳过它或取消导入。
 - 创建完整的文件夹结构
 若选中该选项，则在工作台中创建层次结构以容纳所导入的资源以及文件系统中那些资源的所有父文件夹。
 - 只创建选择的文件夹
 若选中该选项，则在工作台中创建层次结构以容纳所导入的资源。

⑤ 完成上述步骤时单击“完成”按钮。所选文件将导入到工作台，并显示在导航器的目标项目中。

在 Windows 中，可以使用拖放或复制并粘贴操作将文件夹和文件从文件系统复制到工作台中。具体可采用两种方式：

- 打开操作系统的文件系统资源管理器。定位导出的文件，并将它拖到工作台中的导航器视图中的特定位置。

注意：必须将资源拖到想要资源驻留的导航器视图层次结构中的确切位置；不能只是将它们拖放到导航器视图上的空白区。

- 通过在系统的文件资源管理器中复制文件，然后在导航器视图中选择目标并选择“编辑→粘贴”。

除使用“导入”功能之外，将已有的文件加入到项目中的另两种方式是：

- 从 Windows 的资源管理器中直接拖放文件到项目中。
- 直接将文件拷贝到项目文件夹中，然后在 Eclipse 的资源浏览窗口中选择项目或文件夹并执行“刷新”功能。

项目文件夹可以放在计算机的任何位置，并且可以在 Eclipse 中用新建项目的方法将项目路径指定到已经存在的项目文件夹，然后在 Eclipse 中刷新即可。但要注意的是，任何项目文件夹的建立或指定，目前都只能在 Eclipse 中用新建项目的方法来实现，即使是在默认存储项目文件夹的 workspace 路径下新建立一个文件夹，在 Eclipse 环境中也是无法将它变成一个项目，也就是说，这个文件夹对 Eclipse 是不可视的。

2. 导出文件

与导入文件类似，可以使用导出向导从工作台将文件导出至文件系统。

① 在导航器中选择导出文件所在的项目。

② 从菜单栏中选择“文件”→“导出”，或在所选项目的上下文弹出菜单中选择“导入”。

③ 在导出向导中，选择“文件系统”，然后单击“下一步”按钮。

④ 如图 3.24 所示，选择导出文件所在的项目。在右窗格中选择所需导出的文件。也可以使用“选择类型…”按钮来过滤想要导出的资源的类型。

⑤ 在目录字段中，输入或进行浏览以选择想要导出的资源驻留的文件系统中的位

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
第3本教程由站长百科收集整理

置。如果用户输入了不存在的目录名，则当单击“完成”按钮时，就会提供导出向导以创建该目录。



图 3.24　导出资源到本地文件系统

⑥ 在“选项”区域中，可以选择：

- 覆盖现有的资源而不发出警告
 该选项确定导出资源时是否应不进行提示就覆盖已存在于文件系统中的资源。如果此选项处于关闭状态，则在覆盖给定文件之前将会有提示，在这种情况下，开发者可以覆盖该文件、跳过它或取消导出。
- 为文件创建目录
 在文件系统中创建层次（文件夹）结构。
- 创建文件的目录结构
 在文件系统中仅为选择的文件夹创建层次结构。

⑦ 完成上述步骤时单击“完成”按钮。

同导入资源类似，在 Windows 系统中，可以使用拖放或复制粘贴操作将文件夹和文件从工作台复制到文件系统。

3.5.5　删除资源

在 Eclipse 工作台中删除资源非常方便，只需在导航器视图中选择相应的资源，然后可用以下几种方式删除资源：

- 从该资源的弹出菜单中选择删除
- 按 Del 键删除
- 从下拉菜单中选择“编辑”→“删除”

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

Eclipse 工作台将要求用户对删除操作进行确认。

3.5.6 复制、重命名和移动资源

在 Eclipse 工作台中，可以使用导航器中的弹出菜单操作来复制、移动和重命名工作台资源。

在导航器中复制资源的步骤如下。

① 在导航器视图中，选择相应的资源。

② 从该资源的上下文菜单中，选择“复制”（或者 Ctrl+C）。

③ 在导航器视图中，选择目标位置。

④ 从目标的上下文菜单中，选择“粘贴”（或者 Ctrl+V）。

此外，还可以通过在按住 Ctrl 键的同时将文件从一个文件夹拖到另一个文件夹来复制文件。

在导航器中重命名资源的步骤如下。

① 在导航器视图中，选择相应的资源。

② 从该资源的上下文菜单中，选择“重命名”。

③ 导航器将用文本字段来覆盖该资源的名称。输入资源名后按 Enter 键。

在导航器中移动资源的步骤如下：

① 在导航器视图中，选择相应的资源。

② 从该资源的上下文菜单中，选择“移动”，将显示“文件夹选择”对话框。

③ 在“文件夹选择”对话框中，选择目标文件夹并单击“确定”按钮。

3.5.7 资源比较

在 Eclipse 工作台中可以通过比较操作对资源进行比较。资源进行比较操作的步骤如下。

① 在导航器视图中选择一个或多个资源。

② 从资源的弹出菜单中，选择“比较”。

③ 选择一类比较类型，如“互相”等。

比较编辑器将出现在编辑器区域中，如图 3.25 所示。在比较编辑器中，会突出显示文件之间的差异，从而使开发者浏览被比较资源之间的更改。

比较操作的类型有如下几类。

（1）资源库中的最新内容

将选择的资源与当前提交给活动分支的同一资源的版本进行比较。

（2）分支或版本

将选择的资源与已提交给资源库的一个版本或特定分支中的最新版本进行比较。开发者需从列表中选择资源或分支的版本。

（3）补丁程序

显示对资源应用补丁程序所得到的结果。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

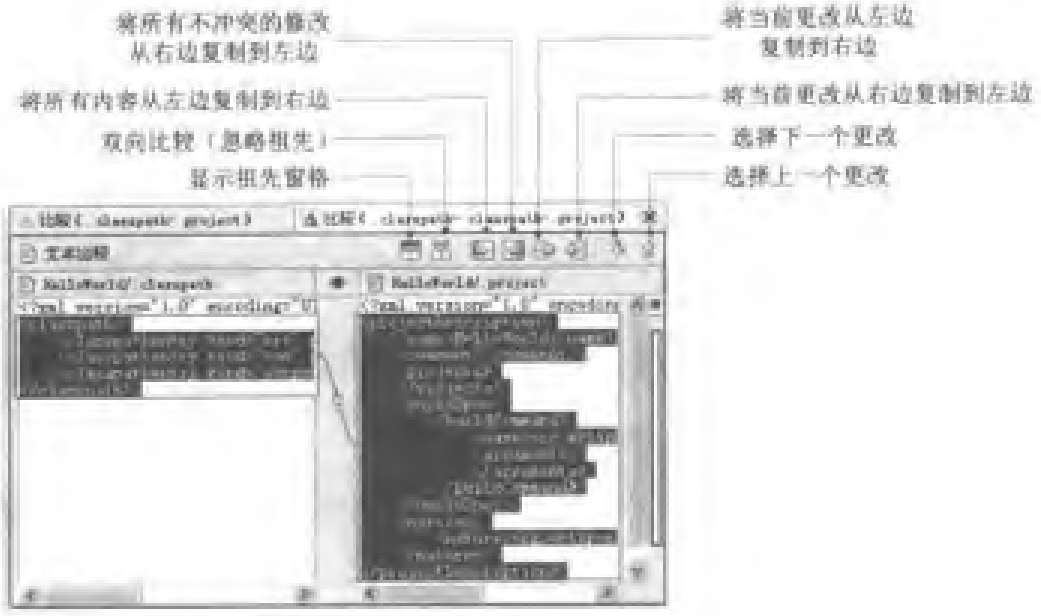


图 3.25　比较编辑器

（4）互相

将选择的两个或三个资源互相进行比较。

（5）修订版

将选择的资源与提交给活动分支的同一资源的版本进行比较。开发者需从列表中选择资源的修订版。

（6）本地历史记录

将选择的资源与本地历史记录中的资源进行比较。

如图 3.25 所示，比较编辑器中的工具按钮提供了各类比较操作，开发者可以根据具体需求选用。此外，开发者可以通过从主菜单栏中选择“窗口”→“首选项”→“工作台”→“比较 / 修订”来定制比较编辑器的行为，如图 3.26 所示。



图 3.26　定制比较操作的首选项

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

比较操作的首选项窗口中的“常规”和“文本比较”选项卡中的选项包括：

- 自动打开结构比较

显示附加信息区域，该区域显示正在比较的资源的基本结构之间的差异。默认情况下，已选中该选项。

- 在状态行中显示附加的比较信息

状态行显示关于比较的附加上下文信息。默认情况下，未选中该选项。

- 忽略空格

比较忽略作为空格字符（空格和制表符等）的差异。还将忽略行终结符之间的差异。默认情况下，未选中该选项。

- 在进行修订之前自动保存脏选项

此选项控制在应用补丁程序之前是否自动保存所有未保存的更改。默认情况下，未选中该选项。

- 在比较查看器中的窗格之间同步滚动

两个比较查看器将互相“锁定滚动”，以保持完全相同，并且每个窗格中的相应代码部分并列排列。默认情况下，已选中该选项。

- 一开始就显示祖先窗格

有时可能想要将资源的两个版本与派生它们的先前版本进行比较。该版本称为它们的公共祖先，并且在三方比较期间，它会出现在它自己的比较窗格中。默认情况下，未选中该选项。

- 显示伪冲突

显示当两个开发者执行相同更改时所发生的冲突，例如，当两个人都添加或除去同一代码行时。默认情况下，未选中该选项。

- 将范围与单行进行连接

控制不同的范围是以可见方式由一条线连接还是由两条线定界的范围连接。默认情况下，已选中该选项。

3.6 任务及任务视图

“任务”视图显示“工作台”中的所有任务和问题，任务视图自带了一组工具按钮，如图 3.27 所示。



图 3.27 任务视图

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

通过“任务”视图，可以查看与特定文件及特定文件中的特定行相关联的任务以及不与任何特定文件相关联的一般任务。无关联的任务不与任何特定资源相关联。

创建无相关的任务的步骤如下。

（1）在“任务”视图中，单击“新建任务”按钮，此时将出现“新建任务”对话框，如图 3.28 所示。



图 3.28　新建任务窗口

（2）在“新建任务”对话框中输入任务的简要描述并按 Enter 键。新任务将出现在任务视图中。如果在输入描述的过程中改变了主意，可按 Esc 键取消对话框。

关联任务与资源中的特定位置相关联，因此要将任务与资源相关联，创建关联任务的步骤如下。

① 在导航器视图中双击关联资源。

② 在编辑器中的任何一行正左方的编辑器区域中，从标记栏中访问上下文菜单。标记栏是编辑区左侧的垂直栏。

③ 从标记栏的上下文菜单中，如图 3.29 所示，选择添加任务。标记栏显示任何标记，包括书签、任务标记（仅适用于相关联任务）和 / 或调试断点。



图 3.29　标记栏的上下文菜单

④ 通过访问紧靠于文件中特定行左侧的标记栏中的上下文菜单，可以将各种标记与该行相关联，并将打开“新建任务”窗口，如图 3.30 所示。

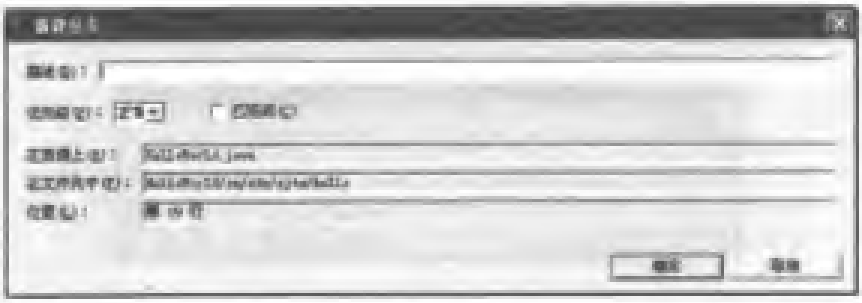


图 3.30　新建任务

⑤ 在“新建任务”窗口的描述字段中，输入想要与文件中的该行相关联的任务的简要描述。

⑥ 单击“确定”按钮完成。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

新的任务标记将出现在标记栏中，处于在其中添加了任务的行的左侧。此外，新任务将出现在任务视图中。

在任务视图中，有两种方法可打开与任务相关联的文件：

① 选择任务，并从上下文菜单中选择“转至”；

② 双击任务。

在这两种情况下，都会打开文件的编辑器，并且会突出显示与所选任务相关联的行。

如需删除任务，则其步骤如下。

① 在任务视图中，访问创建的任务的上下文菜单。

② 在菜单中选择“已完成的标记”，则相应的任务将标记为已完成。

③ 从上下文菜单中选择“删除完成的任务”选项。

完成以上步骤后，任务标记会从标记栏消失，并且会从任务视图中除去该任务。

3.7 书签和书签视图

书签是导航至用户经常使用的资源的简便方法。“书签”视图显示工作台中的所有书签以及与书签相关联文件的名称。可以通过创建书签来将“锚点”置于 Eclipse 工作台中的资源上或置于文件内的特定行处。这样，就可以使用书签视图来快速返回到那些文件。

如在资源透视图中未显示书签视图，则可选择菜单“窗口”→“显示视图”→“书签”，打开“书签”视图。

在 Eclipse 工作台可以为文件或文件中的某些位置添加书签。为文件中的某些位置添加书签，即在文件内创建书签的步骤如下。

① 在编辑器中打开将在某些位置添加书签的文件。

② 将光标定位在文件中的相应行旁的编辑器的标记栏上。

③ 从标记栏上的上下文菜单中，选择“添加书签”选项。

④ 此时将打开“添加书签”对话框，如图 3.31 所示。在对话框中输入对书签的描述。



图 3.31 “添加书签”对话框

⑤ 单击“确定”按钮，新书签将出现在标记栏中，同时新书签还会出现在书签视图中。

类似地，为整个文件添加书签的步骤如下。

① 在导航器视图中，将光标定位到相应的文件。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

② 在文件的上下文菜单中，选择“添加书签”选项。

③ 此时将打开“添加书签”对话框，在对话框中输入对书签的描述。

④ 单击“确定”按钮，新书签将出现在书签视图中。

通过书签可以获取与书签相关联的文件，具体步骤如下。

① 在书签视图中，双击创建的相应的书签。

② 此时将打开文件编辑器，显示与书签相关联的文件。（如果书签对应文件中的特定位置，此时将会突出显示与该书签相关联的行）。

一旦为文件创建书签，即可从“书签”视图快速地打开它们。在书签视图中，还可使用其他两种方法打开与所选书签相关联的文件：

① 从书签的上下文菜单中选择“转至”选项。

② 从视图的工具栏中，单击“转至”工具栏按钮。

在书签视图中，可以很容易地删除所创建的书签。首先选中相应的书签，然后执行下列其中一项操作：

- 单击视图工具栏上的“删除”按钮
- 从书签的上下文菜单中选择“删除”
- 按键盘上的 Delete 键

3.8 定制工作台

在 Eclipse 中可以定制工作台的外观和行为的许多方面，从而可以根据开发者的需求实现个性化定制。

3.8.1 定制快速视图

快速视图是隐藏视图，可以很快使它成为可视视图。它们的工作原理与常规视图相同，但当隐藏它们时，快速视图不会占用工作台窗口上的屏幕空间。

可以用以下两种方法来创建快速视图。

（1）使用拖放操作

定位到相应的视图，单击标题栏并将它拖到窗口最左边的快捷方式栏中。一旦在快捷方式栏上，光标就将更改为叠放光标。释放鼠标按钮，将相应的视图放在快捷方式栏上。快捷方式栏将包括表示该快速视图的按钮。

（2）使用视图菜单中的菜单操作

在要创建快速视图的视图标题栏上弹出上下文菜单，如图 3.32 所示。从此菜单中选择“快速视图”。

例如，如将导航器视图设置为快速视图，则 Eclipse 工作台窗口最左侧的快捷方式栏将出现导航器快速视图的图标，如图 3.33 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理



图 3.32　视图的标题栏上的上下文菜单



图 3.33　快速视图的图标

将视图设置为快速视图，该视图将不会显示在窗口中。如要在窗口中显示该视图，只需在快捷方式栏中，单击相应的快速视图按钮。可以按正常方式使用该快速视图。使用完该快速视图，如要隐藏快速视图，只需单击另一个视图或编辑器，或者单击快速视图的工具栏上的“最小化”按钮。

若要将快速视图转换回常规视图，可以采用以下几种方式：

- 单击快速视图工具栏上的“快速视图”按钮，如图 3.34 所示。



图 3.34　“快速视图”按钮

- 从视图标题栏的上下文菜单中选择“快速视图”选项。
- 从工具栏中拖动快速视图图标，并在“工作台”窗口中的某个位置放下它。

3.8.2　定制工具栏

在 Eclipse 工作台中，工具栏各部分由一条细竖线分隔，可以对工具栏进行定制，重新安排主工具栏的各部分。定制工具栏的步骤如下。

① 确保解锁了工具栏。如果在细的垂直工具栏分隔器旁边具有粗的垂直栏，则工具栏是解锁的。如果它被锁定，则可在主菜单中选择“窗口”→“锁定工具栏”菜单项来解锁工具栏。

② 在需要的段左边的粗垂直线上移动鼠标来抓住要重新安排的工具栏部分。鼠标光标将更改其形状以指示用户可以单击以移动工具栏部分。

③ 单击并按住鼠标左键以抓住工具栏部分。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

④ 将该部分左右或上下移动。释放鼠标键以将其放置在新位置。

⑤ 为了防止对工具栏的意外更改，在主菜单中选择“窗口”→“锁定工具栏”工具栏菜单项再次锁定它。

3.8.3 定制键绑定

在 Eclipse 中可以定制若干键盘的功能，具体方法是在 Eclipse 主菜单中选择“窗口”→“首选项”→“工作台”→“键”，如图 3.35 所示。在“键”首选项页面中，可以为 Eclipse 中的许多命令指定键序列。



图 3.35 “键”首选项

3.8.4 更改字体和颜色

默认情况下，工作台使用操作系统提供的字体。更改这些字体的步骤如下。

① 在 Eclipse 主菜单栏上，选择“窗口”→“首选项”。

② 展开“工作台”类别并选择“字体”项，如图 3.36 所示。

③ 在“字体”列表框中选择想要更改的字体。

④ 单击“更改”按钮。

⑤ 如图 3.37 所示，使用打开的“字体”对话框来选择字体、字形、大小、效果、字符集和颜色等。

⑥ 单击“确定”按钮。

类似地，可设置 Eclipse 中用来显示错误文本和超级链接文本的颜色，其步骤如下。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 3.36 字体定义

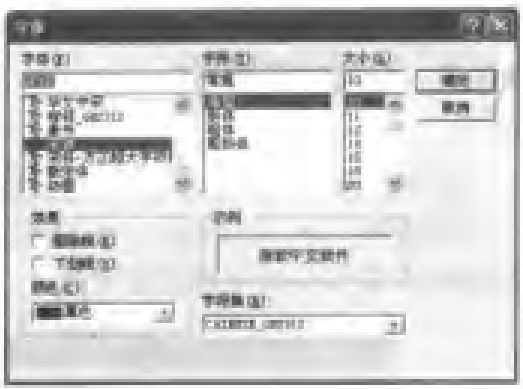


图 3.37 “字体”对话框

① 在 Eclipse 工作台主菜单栏上，单击“窗口”→“首选项”。
② 展开“工作台”类别，并选择“外观”。
③ 选择想要更改的颜色。
④ 使用打开的对话框来选择颜色。
⑤ 单击“确定”按钮。

3.8.5 更改选项卡的放置

除了在工作台中拖放视图之外，还可以重新排列选项卡表示视图的次序，具体步骤如下。

① 单击想要移动的视图的选项卡，并将它拖至期望的位置。当将视图拖到其他视图选项卡上时，就会出现如图 3.38 所示的叠放符号。



图 3.38 选项卡叠放符号

② 当使视图选项卡位于期望位置时，释放鼠标按钮。

经过以上两步即可完成移动所选择的视图。

3.8.6 定制透视图

在 Eclipse 工作台中，可以对透视图进行配置，选择在各个透视图中出现的内容。配置透视图的步骤如下。

① 切换至想要配置的透视图。
② 在 Eclipse 工作台的主菜单中选择“窗口”→“定制透视图”。
③ 此时将打开“定制透视图”对话框，如图 3.39 所示。展开想要定制的项。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 3.39　定制透视图对话框

④ 使用复选框来选择想要在选择的透视图中的下拉菜单上看到哪些元素。未选择的项将仍可以通过单击其他菜单选项来访问。

⑤ 单击“确定”按钮。

经过上述步骤，相应的透视图中将包含所配置的对象。

在 Eclipse 中，用户可以自定义透视图。如果已通过添加、删除或移动视图的方式修改了透视图，则可以保存相应的设置以供将来使用。保存自定义的透视图的步骤如下。

① 切换至想要保存的透视图。

② 在 Eclipse 工作台的主菜单中选择“窗口”→“将透视图另存为”。

③ 此时将显示“将透视图另存为…”对话框，如图 3.40 所示。在名称字段输入透视图的新名称。

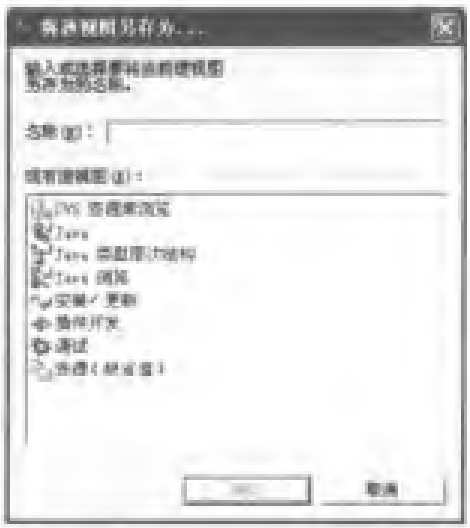


图 3.40　“将透视图另存为…”对话框

④ 单击“确定”按钮。

经过上述步骤，新透视图的名称将添加至“窗口”菜单项下的“打开透视图”子菜

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

单中。类似地，用户也可删除自定义的透视图。删除自定义的透视图的具体步骤如下。

① 在 Eclipse 工作台的主菜单栏上，单击“窗口”→“首选项”。

② 此时将打开“首选项”窗口，展开“工作台”。

③ 单击“透视图”项，如图 3.41 所示。



图 3.41 “透视图”首选项

④ 从“可用透视图”列表，选择想要删除的透视图并单击“删除”按钮。

⑤ 单击“确定”按钮。

需要注意的是，用户仅可以删除自定义的透视图，不能删除 Eclipse 工作台预先定义的透视图。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 4 章　Eclipse 的 Java 集成开发环境

Eclipse 的开发环境并不局限于某种开发工具或开发语言，换句话说，在 Eclipse 中并不存在默认的开发语言。然而，由于 Eclipse 就是使用 Java 编写的，并且 Java 是当前最流行的开发语言，因此到目前为止，Eclipse 对 Java 的开发支持最强。Eclipse 的 Java 集成开发环境具有开发 Java 程序所需的各种强大的功能，Java 开发工具（JDT）实现了全功能的 Java 开发环境。

4.1　验证 JRE 安装和类路径变量

由于 Eclipse 软件开发工具箱中并不包括 Java 开发工具箱（JDK）。因此，在 Eclipse 中使用 Java 进行应用开发，必须首先确认已正确安装 Java 运行时（JRE）。确认检测到 JRE，并确认类路径变量。

默认情况下，用来运行工作台的 JRE 将用来构建和运行 Java 程序。它应与安装的 JRE 列表中的选择标记一起出现。除了项目和 JAR 之外，Java 项目的构建路径还可以包含变量。这些变量称为类路径变量，它们可以指向 JAR 文件。验证步骤如下。

① 通过选择菜单项“窗口→首选项”打开工作台首选项。

② 通过在左边的树窗格中选择“Java→已安装的 JRE”转至安装的 Java 运行时环境首选项页面。如图 4.1 所示。



图 4.1　已安装的 Java 运行时环境

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Java 运行时的定义包含：

- Java 运行时的类型：Java 虚拟机的类型
- Java 运行时的名称：即为此 Java 运行时定义的输入名称
- 安装 Java 运行时的位置：即 Java 运行时的主目录
- Javadoc 的位置（URL）
- 包含 Java 系统类的系统库。如 JAR 文件或源文件

可以安装任意多个不同的 Java 运行时环境。可以切换工作台的默认 Java 运行时。默认的 Java 运行时是将预定义的类路径变量 JRE_LIB、JRE_SRC 和 JRE_SRCROOT 绑定至的 Java 运行时。

然后通过在树窗格中选择工作台来转至工作台首选项页面，确认选择了“资源修改时自动执行构建”选项。

4.2 在 Eclipse 中编写一个简单的 Java 程序

下面，我们首先编写一个简单的“Hello World”Java 程序，讨论 Eclipse 平台的 Java 应用开发。

4.2.1 Java 透视图

在大多数情况下，Java 透视图最适于处理 Java 语言编程元素。该透视图由几个窗格和视图组成，便于执行与 Java 编程相关的任务。

在新的工作空间（或称工作区）创建新的 Java 项目之后，Eclipse 将打开 Java 透视图。在窗格的最左边可看到两个叠在一起的视图：包浏览视图和层次视图。通过单击各自的标签可以在视图间切换。

包浏览视图显示每个项目中的所有包。可以单击包名旁边的加号（+）图标来展开包。在包的名称下面可以找到包中的 Java 类型。类和接口也可以通过单击名称旁边的加号（+）图标进行扩展。扩展 Java 类型时可以看到在其中定义的方法。还可以看到包和导入的声明以及定义的任何变量。

层次视图显示类型的继承链。要激活这个视图，首先在大纲视图（或称轮廓视图）或编辑器中选择一个类型，然后从选择类型的上下文菜单中选择“打开类型层次结构”菜单选项。或者可将元素，如项目、包和类型等，拖放到此视图上。

窗口中部的一大块区域是编辑视图。根据所查看文件的类型，在这个区域中显示不同的编辑器。Java 编辑器是这些专用编辑器中的一种。

该透视图最右边的视图是大纲视图。当在编辑器中打开 Java 类型时，它的层次轮廓就显示在这里。这个视图用于快速浏览类型的不同部分。在这里还有其他可用的功能，如通过抽取代码到新的方法来重建类、重命名包以及修改方法签名。

任务视图主要用于显示工作空间中的错误和警告信息，如句法错误或不能解析的引用。每个条目显示进一步识别该问题的描述和位置。在一条错误信息上双击会打开有问

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

题的文件并将光标定在有问题的行和列上。

4.2.2 创建一个新的 Java 项目

在 Eclipse 中，如果不将包和类与 Java 项目关联起来就不能创建它们。在 Java 项目中，创建的包和类型（类和接口）组成了应用程序。Java 项目在 Java 档案文件（JAR）中可以作为库导出并包含在其他应用程序的类路径中，从而可以使用这些类。创建 Java 项目的步骤如下。

① 首先按第 3 章所述方法，启动 Eclipse，打开新建项目向导。

② 在向导页面的左边窗格中，选择 Java，然后在右边窗格中，选择 Java 项目。

③ 单击“下一步”按钮，在“新建 Java 项目”窗口中填写项目名，本例中项目名为“Hello World Project”。

④ 单击“完成”按钮，在工作台中将打开一个新的 Java 透视图，且新 Java 项目将显示在包浏览器中。如图 4.2 所示，与新的 Java 透视图一起，新的菜单选项和特定于 Java 的按钮被装入到工作台工具栏中。其他按钮和菜单选项将会随活动的视图或编辑器的不同呈现不同的可用性。



图 4.2 Java 透视图中的菜单和按钮

Java 项目具有相关联的 Java 构建器，在更改 Java 源文件时，Java 构建器可以增量编译这些 Java 源文件。

Java 项目还维护有关类型层次结构以及 Java 元素的引用和声明的信息。当用户更改 Java 源代码时，此信息会不断地更新。它与构建器无关。

可以用两种不同方法来组织 Java 项目。

- 将项目用做源容器。对于简单的项目，比较适宜使用此种组织方式。
- 将源文件夹用做源容器。对于较复杂的项目，使用此种组织方式，能将包划分成多个组。

4.2.3 创建一个新的 Java 包

现在，在工作空间上就有了一个 Java 项目。当创建 Java 工具类或独立的 Java 应用程序时，可在 Java 项目中为应用程序创建包和类，可以创建一个或多个 Java 包，包中包含应用程序的类型。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

创建包非常容易，同样有多种方法可以实现。可以使用工具条按钮；或从主菜单条上选择“文件”→“包”；或从 Java 项目的上下文菜单选择“新建”→“包”。无论采用哪种方式，都将打开“新建 Java 包”窗口。

在“新建 Java 包”窗口中，要求输入新的包名。一般地，包名的惯例是使用开发组织的 URL 的逆序再加上描述词。通常以小写字母形式输入包名。图 4.3 显示了这个对话框。



图 4.3　命名新的包

4.2.4　创建一个新的 Java 类

创建包之后，可以继续创建包中包含的 Java 类型。本书中的 Java 类型指的是 Java 类和接口。Eclipse 通常以同样的方式处理它们。在无须区分这两者的地方，统一使用“类型”这一词表示它们。

在工具条上单击“创建 Java 类”按钮或通过包浏览器的上下文菜单，则将显示“新建”窗口，创建新的 Java 类，如图 4.4 所示。



图 4.4　新建 Java 类

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在创建 Java 类的“新建”窗口中，各定义项的含义如下：

- 源文件夹——在“源文件夹”栏中指定类所在的项目，通常与包所在的 Java 项目同名，但可以是工作空间中另一个 Java 源文件夹。
- 包——可在工作空间的任何包中创建类。如果向导启动时已选择了一个包，则包名就自动在这里输入。也可以是新的包名，在这种情况下，Eclipse 将创建该包。假如项目中没有创建包，在此处又没有指定包名，则 Eclipse 将自动创建一个“默认”包。
- 封装类型——这个复选框允许将这个类创建为嵌套类。如果选中该复选框，还需提供该封装类型的名称。
- 名称——在名称栏中输入 Java 类的名称，在本例中为：HelloWorld。类名是必填项，提供了类名后才可以继续，而所有其他的值都是可选的。
- 修饰符——这些复选框和单选按钮允许设置任意类修饰符如 public、final、abstract 或 default。
- 超类——可以输入名字来指定超类，或单击“浏览”按钮生成工作空间所有类的一个列表，从该列表可以选择一个类。
- 接口——单击添加按钮生成工作空间中接口的一个列表，从该列表可选择新建的类必须实现的接口。
- public static void main(String[] args)函数——如果这是一个独立的类，单击该复选框生成 main 方法。在本例中，单击复选框“public static void main(String[] args)”方法，让 Eclipse 创建 main 方法。
- 来自超类的构造器——单击该复选框，向导将生成和超类相匹配的构造函数的代码。
- 继承的 abstract 方法——实现接口时，单击该复选框，向导将为接口上定义的方法生成存根（stub）。

单击“完成”按钮，则包浏览器中出现 HelloWorld.java 文件，单击该文件，编辑器显示该文件。在 main 方法中输入：

```
System.out.println("Hello World");
```

以及注释语句：

```
//This is my first Eclipse App.
```

如图 4.5 所示，在编辑器中，不同的语法部分以不同方式呈示突出显示，包括：

① 常规注释：如//This is my first Eclipse App.

② Javadoc 注释：可通过“窗口→首选项→Java→代码生成”的代码生成选项，对新类型和新文件启用和禁用注释创建。

③ 关键字：如 public、static 等。

④ 字符串。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

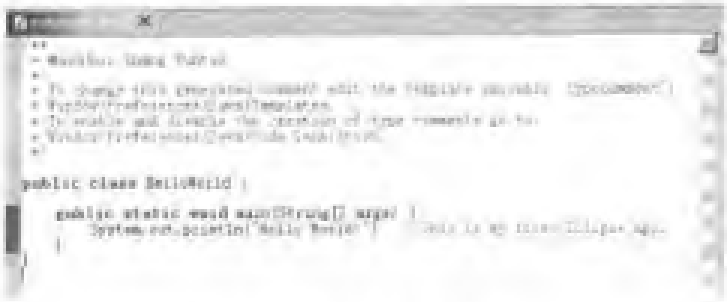


图 4.5　编辑器的显示

保存编辑器内容。Eclipse 有一个增量编译器，因为在工作台首选项页上的默认选择为：资源修改时自动执行构建，所以在每次保存 Java 文件时将进行编译。因此保存编辑器内容时，Eclipse 将自动编译 HelloWorld.java。如果不需要这个功能，可以选择依次选择菜单“窗口”→“首选项”→“工作台”，如图 4.6 所示。



图 4.6　设置自动构建

在窗口中取消“对资源修改执行自动构建”选项，则关闭自动构建特性。

4.2.5　代码编写和代码辅助功能

编辑视图是正在创建的类输入和修改 Java 代码的地方。对于开发 Java 代码，该编辑器提供了许多有效的帮助。

如图 4.5 所示，在这个编辑器中，首先注意到不同的程序元素具有不同的颜色。在类中输入代码时，可注意到缩进是自动实现的，使程序看起来很清晰。如果由于粘贴或移动等原因使代码看起来不太整齐的话，可以让编辑器格式化代码。可从编辑器的上下文菜单选择“源”→“格式化”或按下 Ctrl+Shift+F 组合键来实现格式化。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Java 编辑器使用的一些模板具有代码辅助（也称为内容辅助）功能。代码辅助根据一行中已输入的内容帮助完成该行的代码。可以按 Ctrl+空格键这个组合键激活代码辅助功能。根据光标左边的单词，代码辅助决定如何向开发者提供选择。这些选择显示在弹出窗口上，可以用鼠标也可以用箭头键进行选择。

代码辅助还可用于在代码中插入其他编程元素。例如，定义了一个变量之后，在需要使用该变量的地方输入该变量名的前几个字母，再按 Ctrl+空格键，就会生成一个提供可用术语建议的对话框，从中可以选择该变量。

代码辅助也可用于处理复杂类型的变量名。输入变量名，后面跟句点，然后按 Ctrl+空格键，会生成该变量名表示的类型的可用方法列表。这是非常有用的，因为通常记不清方法名。当带提示的对话框显示时，可以输入方法名的前几个字母，从而去除不包含你输入的字母的那些项，以减少可选择的数目。

可在“窗口”→“首选项”→“Java”→“编辑器”下的“代码辅助”选项卡中查看或设置代码辅助的其他设置，如图 4.7 所示。



图 4.7 编辑代码辅助选项

一个不错的选项是在输入的类名或接口名后的句点后自动给出方法名。可定制激活延迟，默认值为 500 ms。

Java 编辑器与大纲视图相结合。大纲视图显示编辑器中打开文件的有组织的组件视图。根据文件类型，大纲视图会显示不同的东西。例如，如果打开的是 HTML 或 XML 文件，大纲视图显示文件中元素的层次。因此很容易浏览文件的不同部分，例如，到另一张表的一个单元中。

对于 Java 文件，大纲视图显示包和导入语句、类型定义、全局类型变量和方法。选择大纲视图中的这些组件中的任何一个，Java 编辑器将会将光标定位在源文件中所选的元素上。图 4.8 显示了与图 4.5 所示的编辑器内容同步的编辑器内容。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

注意变量和方法名左边的小图标。如“HelloWorldClass()”边上的 C。这些用于给出一个可视的提示，进一步描述正在查看元素的细节。在整个 Eclipse 工作台中有许多这样可视的提示。

在大纲视图中可使用几个过滤器来排序或减少所显示的内容。单击视图标题菜单条下面的四个工具条图标中的一个或多个，可激活过滤器，如图 4.9 所示。



图 4.8　与 Java 编辑器同步的大纲视图



图 4.9　大纲视图过滤器

使用这些过滤器时要小心。可能在过滤后不久，又马上会疑惑所有的静态方法都到哪儿去了。如果认为丢失了类型的一些成员，首先去除所有的过滤器，然后再试一下。

包浏览视图提供了许多与大纲视图相同的功能。主要的差别是包浏览器显示当前工作空间中的所有项目元素，而大纲视图只显示在编辑视图中当前选择的文件细节。

4.2.6　编译和运行程序

在 Eclipse 中，Java 编辑器与编译器相连。编译的默认设置是在 Java 源文件保存时编译。可从主菜单中选择“文件”→“保存”或按 Ctrl+S 组合键，也可单击“保存”按钮进行保存操作。

如果程序出现错误或警告，则错误和警告信息可在以下几个视图中看到。

- 在任务视图中将会生成每个错误的条目。如双击任务，则根据文件类型打开相应的编辑器，并将光标定位在出现错误的位置上。

有时错误描述非常长，不能完全显示在任务视图的空白处。要查看完整的描述，在任务上单击，然后在 Eclipse 窗口底部的信息行阅读错误描述。

- 在包浏览器和大纲视图中的错误以包含白色的×的红圈来表示。类级上的错误表示类中的方法出错或者是类定义本身有错误。错误还在包和导入语句中报告。

警告也以类似的方式报告，但不是以红圈显示，而是以包含黑色感叹号的黄色三角形来表示。

- 编辑器视图的标记条，即编辑器左边的黑色垂直工具条，可以根据情况显示几个图标或标记。包含错误的代码行将显示错误和警告信息。鼠标放在标记上面，将显示与该标记相关的描述。

在某些情况下，会看到在错误标记上面有个像亮着的灯泡一样的标记；这是快速修复标记，这表示 Eclipse 对如何修复该错误有个建议。单击该标记将生成该问题的可能解决方案列表。选择其中一个建议将修复代码并删除快速修复标记，但错误标记直到保存

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

方法时才消失。

快速修复标记可以单独显示，无须错误标记。这种情况说明 Eclipse 已经检测到能提供一些帮助的状态。单击标记会提供以后如何避免错误的建议。

在 Java 视图中，工具栏中有两个按钮，分别用来进行调试和运行。在工具条上单击“运行”按钮，则控制台显示运行结果“HelloWorld”。另外，在包浏览器视图中选择 Java 程序 main 类，单击工具条上“调试”或“运行”按钮，可以方便地运行或调试 Java 程序。

Eclipse 存在多种运行 / 调试程序的方式，为了确定当前项目用哪一种方式运行，需要单击“运行”或“调试”按钮旁的下拉箭头进行选择。通常，程序“运行为”或“调试为”“Java 应用程序”。在这种方式下，如果当前位置是包含 main()方法的 Java 程序，单击“调试 / 运行”按钮就会立即开始执行调试 / 运行功能。如果当前位置是在包或项目上，Eclipse 会搜索出当前位置所包含的所有可执行程序，然后由程序员自己选择运行哪一个。一旦运行程序，它将全速执行。在调试程序时，可通过使用调试器来控制它的运行，用户可以暂停线程，使用断点及检查程序内部。

如需在 Eclipse 环境之外运行 Java 程序，则需将项目导出为包含程序的 JAR 文件，其步骤如下。

① 选择“HelloWorld”项目，并在它的上下文弹出式菜单中选择“导出”。

② 在“导出”窗口中，选择“JAR 文件”选项，并单击“下一步”按钮。

③ 此时将显示“JAR 导出”窗口，如图 4.10 所示。在此窗口中选择导出项目“HelloWorld”，并输入 JAR 文件名，然后单击“下一步”按钮。



图 4.10 设置 JAR 文件导出选项

④ 此时将打开“JAR 封装”窗口，接受默认选项，单击“下一步”按钮。

⑤ 如果从 JAR 文件中的类运行程序，需要指定 JAR 文件中的一个类作为应用程序

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

的入口点（该类包含 main()方法）。单击“Main 类”入口域边上的“浏览”按钮，从列表中选择 cn.edu.sjtu.hello.HelloWorld，如图 4.11 所示。然后单击“完成”按钮。



图 4.11　指定应用程序的入口点

⑥ 进入命令行，执行“HelloWorld”程序：

```
java -jar HelloWorld.jar
```

控制台输出如图 4.12 所示。



图 4.12　在 Eclipse 外运行程序

4.3　Eclipse 的定制

Eclipse 具有灵活的配置。无论是界面的显示，还是应用程序的构建等，开发者可以

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

根据具体的需求进行定制。

4.3.1　首选项

在定制 Eclipse 中使用最多的就是“首选项”页面。首选项的大部分功能是由它们的首选项条目定义的，但是在“首选项”对话框提供两个公用“按钮”——“导入”按钮和“导出”按钮，它们的功能分别是：

- “导入”操作将在首选项中应用用户定义的文件中的设置；
- “导出”操作将把对默认首选项的所有更改写入用户指定的文件。

使用首选项的导入、导出操作，可备份和恢复首选项的设置，并可在不同的 Eclipse 开发环境中共用这些设置。

工作台实现了通用的首选项体系结构。开发者开发的插件可向首选项页面中添加各自的条目。

在前面的章节中，已经介绍了一些首选项条目的设置，下面我们将介绍其他的一些常用的首选项条目的设置。

（1）定义在构建期间不应复制至输出文件夹的资源和资源类型

如果 Java 项目正在使用源文件夹，则在编译项目中的源文件的过程中，Java 编译器还会将非 Java 资源复制到输出文件夹中，因此，在正在运行的程序的类路径上会提供这些资源。

为了避免将某些资源复制到输出位置，可以在编译器首选项中的“构建路径”选项卡下的“已过滤的资源”文本框中设置资源过滤器。如需过滤多个资源或资源类型，可用逗号分隔。

（2）禁用“自动编译”，只在需要编译时进行手工编译

清除“工作台”首选项中的“对资源修改执行自动构建”复选框（默认值是选中的），即可禁用 Eclipse 工作台的“自动编译”功能。

当禁用“自动编译”进行手工构建时，若选中工作台首选项中的“在手工构建前自动保存所有修改过的资源”复选框（默认值是禁用的），则在构建前将保存所有资源。

当需要进行手工构建时，可按 Ctrl+B 键，或者从菜单栏中选择“项目”→“全部构建”。

（3）在大纲等视图中显示方法返回类型

“大纲”等 Java 视图提供了一些方法，如需要显示它们的返回类型。可在首选项页面中导航到“Java”下的“外观”条目，并选择“显示方法返回类型”复选框（默认值是禁用的）。

（4）显式定义构建项目所采用的次序

默认情况下，工作台通过将项目引用解释为先决条件关系来估计构建次序。然而，可以显式定义构建项目所采用的次序，其步骤如下。

① 选中首选项页面中的“构建次序”条目。

② 在“构建次序”首选项页面禁用“使用缺省构建次序”选项。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

③（可选）单击“添加项目”和“除去项目”按钮来将项目添加至列表，或者从列表中除去项目。

④ 选择列表中的一个或多个项目，并单击“向上”或“向下”按钮来设置首选项目构建次序。

⑤ 单击“确定”按钮。

（5）定义类路径变量

在工作台中，类路径变量是全局存储的。换句话说，工作台中的所有项目可以共享这些类路径变量。添加或更改类路径变量的步骤如下。

① 选中首选项页面中的“Java”类别下的“类路径变量”条目。

② 若要添加新的类路径变量，单击“新建”按钮，这将打开“新建变量条目”页面。

若要编辑现有类路径变量，在已定义的类路径变量列表中选择该变量，并单击“编辑”按钮，这将打开“编辑变量条目”页面。

③ 无论在“新建变量条目”页面中，还是在“编辑变量条目”页面中，在“名称”字段中输入变量的名称；在“路径”字段中输入变量要引用的路径。

④ 单击“确定”按钮。

（6）更改选项卡的放置

叠放的视图或编辑器的选项卡可以出现在包含它们的区域的顶部或底部，更改选项卡的放置位置的步骤如下。

① 选中首选项页面中的“工作台”类别下的“外观”条目。

② 使用“编辑器选项卡位置”组或“视图选项卡位置”组中的单选按钮，控制要将这些选项卡位于顶部还是底部。

③ 单击“应用”或“确定”按钮。

（7）定义 import 语句的排序次序

定义 import 语句的排序次序的步骤如下。

① 选中首选项页面中的“Java 类别”下的“组织导入”条目。

②（可选）在“组织导入”可以单击“新建”按钮添加新的前缀；或单击“编辑”按钮更改现有前缀的名称；或单击“除去”按钮从列表中除去所选前缀。

③ 使用“向上”和“向下”按钮以通过将所选前缀上下移动来重新安排列表的顺序。

④（可选）单击“装入”按钮和“保存”按钮从文件中装入前缀列表或者将前缀列表存储到文件中。

⑤ 完成时单击“确定”按钮。

4.3.2 定制 JRE

在 Eclipse 中，可以安装任意多个不同的“Java 运行时环境（JRE）”。JRE 的定义包含：

- JRE 的类型
- 名称

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 安装 JRE 的位置
- Javadoc 的位置（URL）
- 包含 Java 系统类（如 java.lang.Object）的系统库。

除非特地覆盖默认的 JRE，否则，在所有项目中都使用默认的 JRE 来编译和启动 Java 程序。默认的 JRE 是将 JRE_LIB、JRE_SRC 和 JRE_SRCROOT 绑定至的已安装 JRE。如果已经从 JRE_LIB 变量的构建路径中除去了该变量，则不会对默认的 JRE 编译项目。如果为默认的 JRE 的项目设置了定制运行时 JRE，则不会使用默认的 JRE 来启动程序。更改默认的 JRE 的步骤如下。

① 从工作台的菜单栏中，选择“窗口”→“首选项”。

② 选中左窗格中的“Java”类别下的“已安装的 JRE”条目。

③ 选择想要指定为工作台中默认 JRE 的对应行上的框。如果列表中未出现想要指定为缺省 JRE 的 JRE，则必须添加它。

④ 单击“确定”按钮。

在首选项页面中还可修改除 JRE 类型之外的 JRE 定义的所有设置，编辑 JRE 定义的步骤如下。

① 从工作台的菜单栏中，选择“窗口”→“首选项”。

② 选中左窗格中的“Java”类别下的“已安装的 JRE”条目。

③ 选择想要编辑的 JRE 定义，并单击“编辑”按钮，这将打开“编辑 JRE”页面，如图 4.13 所示。



图 4.13 编辑 JRE

④ 在“JRE 名称”字段中，编辑 JRE 定义的名称。相同类型的所有 JRE 都必须具有惟一名称。

⑤ 在“JRE 主目录”字段中，编辑或者单击“浏览”按钮来选择指向 JRE 安装的根目录的路径，通常该目录包含 JRE 的 bin 和 lib 目录。

⑥ 在“Javadoc URL”字段中，编辑或者单击“浏览”按钮来选择 URL 位置。Javadoc 导出向导将该位置用做默认值，“打开外部 Javadoc”操作也使用该位置。

⑦ 如果想要对此 JRE 使用缺省库和源文件，则选择“使用缺省系统库”复选框。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

否则，清除该复选框，并按需要进行定制。也可以对所引用的 jar 文件单击“连接源”按钮。

⑧ 完成时单击“确定”按钮。

4.3.3 格式化 Java 代码

在 Eclipse 中，可以设置 Java 编辑器中 Java 代码的格式化选项，设置步骤如下。

① 从工作台菜单栏中，选择“窗口”→“首选项”。

② 在首选项页面的左窗格中，选中“Java”类别下的“代码格式化程序”条目，此时将打开“代码格式化程序”首选项页面，如图 4.14 所示。



图 4.14 Java 代码格式化程序

③ 在“新行”、“行分割”和“样式”选项卡中，选择想要格式化程序遵循的代码格式化约定。

a）在“新行”选项卡下：

i.“在左花括号之前插入新行”复选框表示编辑器在打开新的花括号之前插入换行符。换句话说，打开花括号始终从新行的开头开始，其默认值是禁用。

ii.“在控制语句中插入新行”复选框表示编辑器在新的控制语句之前插入换行符。换句话说，控制语句（例如，if、else、catch、finally 等）始终从新行的开头开始，其默认值是禁用。

iii.“清除所有空行”复选框表示编辑器将删除文件中的所有空行，其默认值是禁用。

iv.“在‘else if’之间插入新行”复选框表示编辑器在 else-if 语句的词语“else”和“if”之间插入新行，其默认值是禁用。

v.“在空的块内部插入新行”复选框表示编辑器在空花括号之间插入换行符。换句话说，空的花括号集合中的左花括号和右花括号将总是出现在不同的行上。如果右花括号后跟关键字则例外，在此情况下，两个花括号将出现在同一行上，其默认值是启用。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

b）在“行分割”选项卡下：

i.“最大行长度”字段表示任何一行的最大长度，长度比此数目长的行将会折行，在此处输入 0 将完全禁用分行，其默认值是 80。

c）在“样式”选项卡下：

i.“紧凑赋值”复选框表示编辑器将除去变量与赋值语句之间的所有空格以便使它们不对称（例如，a=b;），其默认值是禁用。

ii.“在强制转型后插入空格”复选框表示编辑器在强制转型与跟着的表达式之间插入空格，其默认值是启用。

iii.“插入制表符而不是空格来进行缩进”复选框表示编辑器使用制表符而不是空格来表示缩进，其默认值是启用。

iv.“表示缩进级别的空格数”字段表示如果编辑器使用空格而不是制表符来表示缩进，则这是组成单个缩进的空格数目，其默认值是 4。

v.“预览窗格”显示一个示例来说明，在使用当前显示在此页上的设置的情况下，开发者的 Java 代码看起来会是什么样子。

④ 注意，在该页面的底部，可以观察每个个别代码格式化选项的示范效果，并查看已格式化的代码的预览效果。

⑤ 完成时单击“确定”按钮。

如果按团队形式进行工作，需要团队有统一的代码格式。使用格式化可使代码具有统一的风格。

为了遵循所定的代码标准，还可以定制另一个元素，即创建类型时插入的注释。称为类型注释（typecomment）的模板控制这些类型注释的格式和内容。

可通过“窗口”→“首选项”→“Java”→“代码生成”，然后在“代码生成”窗格中选择“代码和注释”选项卡，如图 4.15 所示。选择“注释”和“代码”树下的某一元素并单击“编辑”按钮可修改模板。



图 4.15 设置注释

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

4.4 其他的与 Java 编程紧密相关的透视图和视图

Java 开发环境插件为 Java 编程提供了许多相关的视图和透视图。

4.4.1 包资源管理器视图

在默认情况下，包资源管理器视图显示在 Java 透视图中。包资源管理器视图显示工作台中的 Java 项目的 Java 元素层次结构。包资源管理器将所有的 Java 元素组织成一棵树，这棵树由项目、包、编译单元和类型等组成。与“导航器”视图中显示的资源对应的 Java 程序的内容将显示在包资源管理器中。元素层次结构是从项目的构建路径中派生出来的。

在包资源管理器视图中，对于每个项目，其源文件夹和引用的库都显示在树中。可以打开并浏览内部和外部 JAR 文件的内容。在 JAR 内打开 Java 元素会打开 CLASS 文件编辑器，并且如果存在 JAR 文件的源连接，会显示它的相应源。

可以使用包资源管理器来浏览 Java 元素，并可以使用过滤器来控制将哪些文件显示在包资源管理器视图中，过滤所显示的元素的步骤如下。

① 在“包资源管理器”视图的工具栏上，单击“菜单”按钮并选择“过滤器”。

② 选择或清除想要应用于视图的过滤器，如：

a）在排除列表中，清除“.*文件”的复选框则显示系统文件；选中“.*文件”的复选框则隐藏系统文件。

b）在排除列表中，清除“Java 文件”的复选框则显示显示 Java 文件；选中“Java 文件”的复选框则隐藏 Java 文件。

c）在排除列表中，清除“非 Java 元素”的复选框则显示非 Java 元素；选中“非 Java 元素”的复选框则隐藏非 Java 元素。

③ 可以选中对话框顶部的“名称过滤模式”复选框选择，并在相应的字段中确定过滤视图的模式，模式将隐藏与模式相匹配的名称，可用逗号分隔多个模式。

4.4.2 层次结构视图

开发者使用层次结构视图来查看类型的完整层次结构，或只查看它的子类型，或只查看它的超类型。类型层次结构视图由两个窗格组成。

① “类型层次结构”树窗格。

② 成员列表窗格（可选），成员列表窗格显示在类型层次结构树窗格中当前所选择的类型的成员。

层次结构视图提供了三种不同的方式来查看类型层次结构，并可使用工具栏按钮在三种方式之间进行切换：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

（1）显示类型层次结构

该方式显示类型的类型层次结构。层次结构视图将显示所选类型的所有超类型和子类型。类型 java.lang.Object 显示在左上角。

（2）显示超类型层次结构

该方式显示类型的超类型层次结构。此视图显示所有超类型和所有实现接口的层次结构。所选类型始终显示在左上角。

（3）显示子类型层次结构

该方式显示类型的子类型层次结构。此视图显示所选类型的所有子类型和所选接口的所有实现者（如果在接口上打开了该视图的话）。所选类型始终显示在左上角。

有多种方式可打开类型层次视图、显示所选对象上的类型层次结构。如要对当前文本选择打开类型层次结构，则在编辑器中选择 Java 元素的名称，并执行下列操作之一。

- 按 F4 键
- 从编辑器的弹出菜单中选择“打开类型层次结构”
- 从菜单栏中选择“浏览”→“打开类型层次结构”

4.4.3 控制台视图

控制台视图显示进程的输出，并可以为进程提供键盘输入。控制台显示三种不同的文本，每种文本都采用不同的颜色。

① 标准输出。默认情况下，应用程序写至标准输出流的文本的颜色是蓝色。

② 标准错误。默认情况下，应用程序写至标准输出流的文本的颜色是红色。

③ 标准输入。默认情况下，输入到控制台中由应用程序读取的文本的颜色是绿色。

在 Eclipse 工作台的菜单栏上选择“窗口”→“首选项”，然后在首选项页面左侧窗格中选择“调试”条目，如图 4.16 所示。

在控制台页面中，单击颜色按钮，将显示“颜色”窗口，可以为这些不同类型的文本选择不同的颜色，如图 4.17 所示。

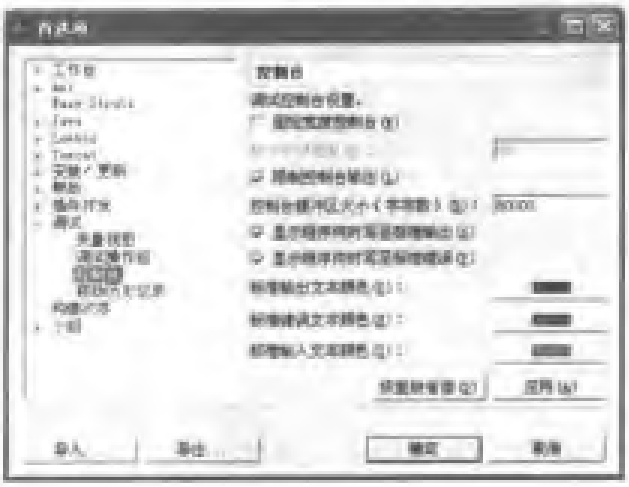


图 4.16 控制台首选项

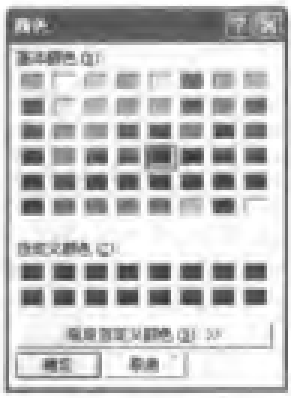


图 4.17 选择文本颜色

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在图 4.16 中，当选中“显示程序何时写至标准输出”复选框，则每当将新输出从程序的标准输出流写入控制台时，将显示控制台视图。当选择“显示程序何时写至标准错误”复选框，则每当将新输出从程序的标准错误流写入控制台时，将显示视控制台视图。

在图 4.16 中：

- “固定宽度控制台”复选框表示了控制台是否具有固定字符宽度。选中该复选框时，必须指定最大字符宽度。某些应用程序将很长的行写至控制台，需要进行水平滚动才能阅读这些行。可通过将控制台设置为使用固定宽度从而自动回绕控制台输出来避免发生此情况。该复选框的默认值是禁用。
- “限制控制台输出”复选框表示了在控制台中缓冲的字符的数目。当打开时，还必须指定最大缓冲区大小。当控制台输出超出指定的大小时，在缓冲区的开始处将输出截断。最大缓冲区大小的默认值是 80 000 字节。“限制控制台输出”复选框的默认值是启用。

控制台上下文菜单提供了操纵视图中的文本所需的各种操作。“全部选中”命令选择控制台视图的当前窗格中的所有文本。“复制”命令将选择的所有文本从控制台视图复制到剪贴板上。“清除”命令清除控制台视图中的所有内容。

4.4.4 浏览透视图

Java 浏览透视图是用于浏览 Java 项目结构的透视图。它包含一个编辑器区域和下列视图：

- 项目
- 包
- 类型
- 成员

在 Eclipse 工作台的菜单栏中，选择“窗口”→“打开透视图”→“Java 浏览”将打开 Java 浏览透视图，如图 4.18 所示。



图 4.18 Java 浏览透视图

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

如图 4.18 所示，浏览透视图使用不同的视图来显示相同的信息。在一个视图中选择元素将在另一个视图中显示该元素的内容。透视图的视图通过下列方式彼此链接：

- 在项目视图中选择元素，则它的包显示在包视图中。
- 类型视图将显示包视图中选择的包所包含的类型。
- 成员视图显示所选类型的成员。成员视图在功能上与普通 Java 透视图中使用的大纲视图相当。
- 在成员视图中选择元素将在编辑器中显示该元素。如果没有为该元素打开编辑器，则双击该元素将打开相应的编辑器。

默认情况下，四个视图全都链接至活动编辑器。这意味着这些视图将根据活动编辑器中显示的文件来调整它们的内容和选择。

4.4.5 Java 类型层次结构透视图

Java 类型层次结构透视图是用于浏览类型层次结构的透视图。可对类型、编译单元、包、项目或源文件夹打开该透视图，它由层次结构视图和编辑器组成。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 5 章　Eclipse 中的 Java 程序调试

本章将讨论在 Eclipse 中查找程序中的逻辑问题所需的基本调试技术。在程序开发过程中总会碰到若干错误。这些错误导致应用程序不能按设想的模式执行。即使程序语法正确，能够编译，但在运行时仍可能产生不可预计的异常（exception），或产生错误的结果。

Eclipse 有个优秀的内置源代码调试器，具有集成开发环境所应具有的所有调试功能。调试器透视图提供快速查找错误所需的所有视图。

5.1　概述

使用 Eclipse 开发应用程序，在修改或增加新代码后保存文件时，编译器会发现一些错误，这些错误容易修复。更严重的问题在程序编译甚至运行时发生，如生成的结果不是所期待的结果，或显示一个错误的值等。这时就体现出一个优秀的调试器的价值了。Eclipse 提供了一个强大的调试透视图。在该透视图中可执行任何和调试有关的操作。

5.2　调试透视图简介

在 Eclipse 中调试程序，所需做的只是单击“调试”按钮，如图 5.1 所示。



图 5.1　调试和运行工具条图标

第一次调试 Java 程序时，需要定义程序所需的运行时环境类型。如果没有所需的特殊的参数或设置，可在工具条按钮中直接选择“调试为”→“Java 应用程序”，如图 5.2 所示。

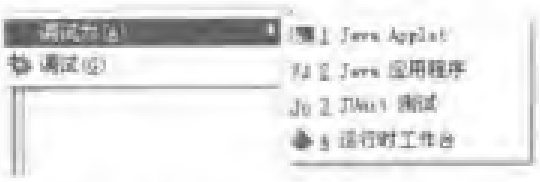


图 5.2　“调试为”菜单选项

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

如果需要定制调试程序以提供命令行参数，选择不同的 JRE 或其他参数，从图 5.2 所示的工具按钮选项中选择“调试”，将显示创建、管理和运行配置的调试对话框，如图 5.3 所示。



图 5.3 创建、管理和运行配置

5.3 常规调试技术

如果没有设置断点，程序将一直运行到结束或直到遇到一个运行时错误。可在怀疑导致问题的地方或之前一点设置断点。一旦运行到断点，就单步执行并检查变量，识别并修复错误，重新测试，然后继续执行直至结束。

5.3.1 断点

在代码中设置断点，以便调试器能够挂起执行的程序，让程序员查找错误原因。断点可在注释行之外的真正的 Java 代码的任意一行设置。断点让调试器在 Java 程序执行到断点时挂起。

当挂起执行时，调试器获取控制权并提供对程序各部分的访问，如变量值和调用栈的当前状态，此外还提供更细的控制程序执行的方法，如步进程序的能力。

可在 Java 编辑器视图设置行断点。在查看 Java 文件时，将鼠标放在设置断点的代码行的标记条上。标记条是位于 Java 编辑器视图最左边空白处的灰色垂直标记条。双击语句左边的灰色区域，则断点将会添加到该语句中。

有两种方式设置断点：

① 双击代码旁的标记条，在设置断点和重置断点之间切换。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

② 在代码旁的标记条处单击右键，从上下文菜单中选择“添加 / 除去断点”。

断点在标记条上以蓝色的圈表示。一旦在虚拟机上载入了类并安装了断点，纯蓝色的圈就变为蓝色的圈加上一个勾。标记条上的白色圈表示禁用断点。通常程序一直运行到一个启用的断点，此时程序挂起并打开调试透视图。

可打开包含断点的源代码文件，在标记条上查找蓝色圈来查看断点。虽然这种方法可以快速地查看程序下一步会停在何处，但不是查看所有断点的好方法。要查看工作空间中文件（其中有些文件可能当前在 Java 编辑器中没有打开）代码的所有断点，可以使用断点视图。

默认情况下，断点视图位于调试透视图的右上角的窗格中。在这个视图中可以查看当前在工作空间中设置的所有断点。在该视图中还可以通过双击断点来查看在源代码中的对应行。还可在该视图上通过每个断点的上下文菜单来操纵断点，如图 5.4 所示。

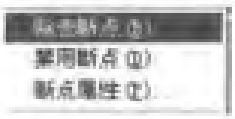


图 5.4　断点上下文菜单

5.3.2　调试透视图

在 Eclipse 中调试应用程序，一旦到达一个启用的断点，执行将会挂起，并且 Eclipse 工作台将打开调试透视图。

调试透视图提供一个预先设置的窗口，如图 5.5 所示，包括几个特定的窗格和视图，它们有助于判断代码中哪里出现了错误。

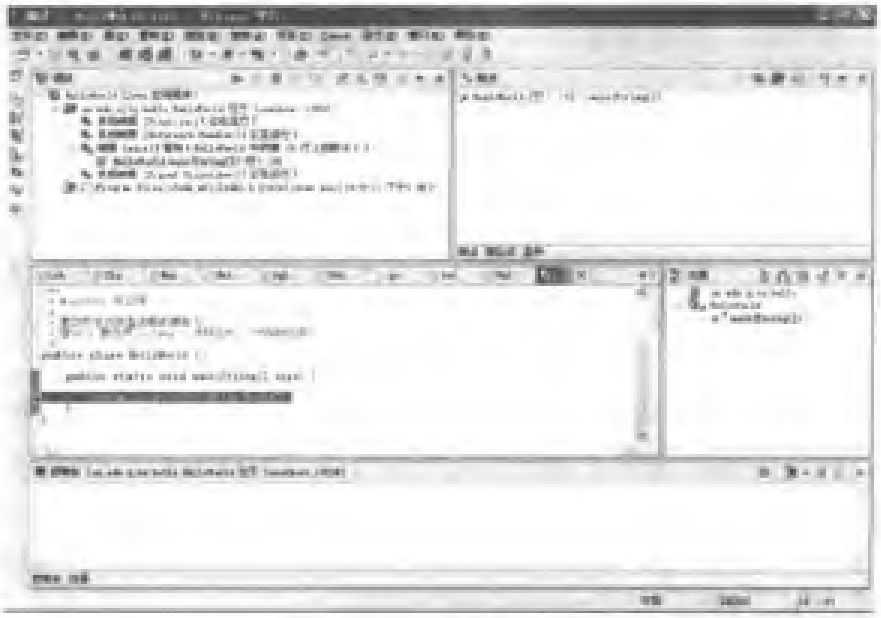


图 5.5　调试透视图

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

图 5.5 显示一个非常典型的透视图界面。下面将简要地介绍透视图中的不同视图。

在图 5.5 所示的透视图中，底部的窗格包含堆叠的控制台和任务视图。在 Java 透视图中，它们的功能一样。程序在调试器控制下运行时，控制台视图显示 System.out.println() 语句生成的任何调试信息。错误信息也将显示在控制台上。

左上角的窗格为调试视图。可在调试视图中查看当前运行的进程和线程。在这里也显示任何被终止的进程。从视图的上下文菜单中选择“除去所有终止的启动”可删除被终止的进程，如图 5.6 所示。这个上下文菜单可以控制正在运行的、挂起的或终止的进程。



图 5.6 调试视图的上下文菜单

当在调试器中选择一个调试选项来运行程序时，将打开调试透视图，如图 5.7 所示，调试视图将高亮显示程序执行挂起处的进程和线程。

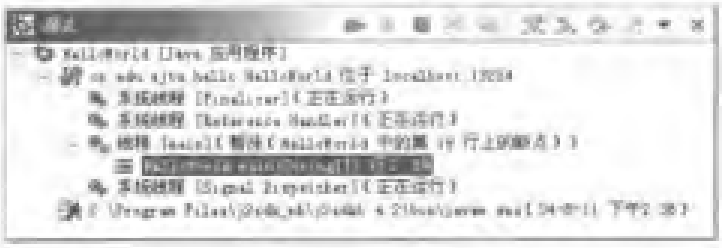


图 5.7 调试视图

在图 5.7 中，显示了一个进程 HelloWorld，这个正在运行的类有几个线程。并不是所有的线程都由程序员创建的，其中有一些是系统线程。三个线程正在运行，一个线程挂起。绿色的三角形图标表示正在运行的线程，两个垂直条图标表示一个挂起的或暂停的线程。

编辑器视图包含正在调试的源代码文件。在源代码文件中，高亮显示将要执行的下一行。编辑视图与调试视图中选择的栈框架同步。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

大纲视图与栈视图中的当前选择同步。同时，在大纲视图中选择的任何元素在源代码编辑器视图中也将被选中。

大纲视图和编辑器视图在调试透视图中的操作与在 Java 透视图中完全一样。但当代码在断点处挂起时，编辑器窗格中还有一些其他的功能。

- 将鼠标覆盖在变量名上会显示有关该变量的信息。变量的类型和当前值显示在弹出窗口上。复杂对象显示其对象 ID。
- 可直接从编辑器窗格选择并计算 Java 表达式。如图 5.8 所示，选择表达式，并从上下文菜单中选择“显示”，可以查看表达式的结果。

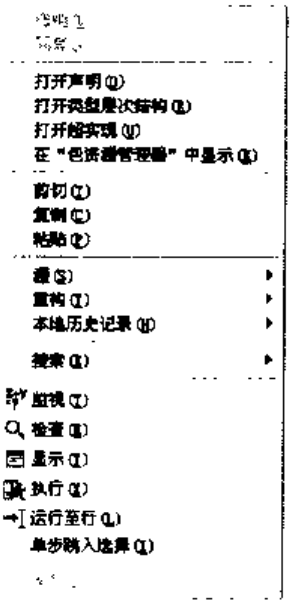


图 5.8　显示表达式

检查或显示表达式的结果显示在显示视图中。显示视图位于调试透视图右上角的窗口中。

5.3.3　控制程序执行

Eclipse 存在多种运行和调试程序的方式。为了确定当前项目使用哪一种方式运行，需要单击“调试”按钮的下拉箭头进行选择。选中 Java 应用程序，然后单击“调试”按钮，程序将运行到设置的断点处，然后停止运行。按 F6 键，程序将单步执行，这时程序运行到下一条语句。如按 F8 键，程序将运行到程序结束。

控制程序执行的另一个方法是在程序出错前设置断点以挂起应用，然后步进程序，更仔细地查看程序的流程并在程序缓慢向出错点推进时检查变量的情况。工具条按钮和功能键提供了步进和线程控制，如图 5.9 所示。

步进程序时有以下三种不同的控制方法。

① 如果下一步要执行的是方法调用，“单步跳入”将进入调用的方法内。这个功能也可以通过 F5 功能键实现。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 5.9 调试工具条按钮

② “单步跳过”不进入调用的方法内部；而是执行方法并返回到方法调用后的下一步。这个功能还可以通过 F6 功能键实现。

③ 如果已经步进了方法调用，并已看到希望看到的，希望从该方法快速返回，则选择“单步返回”。这个功能还可以通过 F7 功能键实现。

如果在没有源代码的情况下运行步进方法代码，则会产生一个对话框，在对话框中可输入源代码的位置。在使用只包含类文件的库代码时会发生这种情况。在步进 Java 类库代码时也会经常发生这种情况。可以通过设置步进过滤器来避免进入这种打乱思路、不方便的情况。过滤器中的任何包或类在步进入只包含二进制的库时，调试器将不弹出要求指定源代码位置的对话框。调试器不进入定义为步进过滤器的包或类。

要设置步进过滤器，选择“窗口”→“首选项”→“Java”→“单步过滤”。如图 5.10 所示，可以将二进制包作为默认值预先定义。可从调试视图的上下文菜单中将步进过滤器设置为开或关。



图 5.10 调试步进过滤

除了刚才提到的步进控制，调试工具条还包含管理进程和线程的其他三种控制。

- “继续”（Resume）功能用于启动一个挂起的线程。在调试了一部分代码后，如想快速继续到下一个断点或如果没有更多的断点则直接将程序运行到结束，则可使用这一功能。这个功能还可通过 F8 功能键实现。
- “暂停”（Pause）功能用于挂起一个当前正在运行的线程。选择正在运行的线程并

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

单击 Pause 后，该线程在下一步将挂起。

- “终止”（Terminate）功能用于终止一个线程。使用这个功能可从无穷循环或挂起的状态退出或强制退出应用程序。

另一种在代码中快速移动的方法是使用“运行至行”（Run to Line）方式，这种方式可在调试透视图中的 Java 编辑器视图的上下文菜单中选择。这种方式可以运行到光标所在的行。程序从它当前的位置开始运行一直运行到光标所在处，然后程序挂起。在执行“运行至行”期间遇到的任何断点都将被处理。

5.3.4 检查程序的执行情况

Eclipse 的日志文件保存在项目的工作空间 workspace/.metadata/.log 文件中，这个文件用来保存运行时的错误，其中的信息对于调试程序和排除错误非常有用。

5.4 高级调试技术

5.4.1 表达式的计算和显示

表达式视图显示正在检查的变量。当应用程序处于挂起状态时，如在断点处停止或步进代码时，可以检查作用域内的任何变量。默认情况下，表达式视图位于透视图窗口的右上角，它的窗格与断点视图、变量视图等一些其他视图共享。

检查变量可以查看复杂变量各部分的值。表达式视图初始是空的，可以将感兴趣的变量加进去。表达式视图中的变量在上下文中仍然适用，除非使用视图中的工具条按钮将它们删除。

显示视图提供许多计算 Java 表达式的方法。这是 Eclipse 调试透视图的一个非常强大的功能。显示视图提供进入正在运行的 Java 虚拟机的窗口，当调试器挂起应用程序时，可使用该窗口访问 Java 虚拟机。

在显示视图中，可遵循以下步骤计算表达式：

- 在显示视图中输入表达式；
- 用鼠标选中表达式，高亮显示它；
- 从上下文菜单中选择 Display。

计算结果就显示在表达式下面。

显示视图中执行的代码在当前栈框架的上下文中运行。代表类实例的复杂变量可调用它们的方法。这是非常有用的，因为这样可以调用不属于程序的方法。如果该方法返回一个值而不是 void，则该值在执行后会显示出来。切换到变量视图可确认确实运行了方法。

在显示视图中输入代码时，可使用代码辅助（Ctrl+空格）来帮助选择正在使用的类的正确成员。在显示视图中不能声明新的变量。显示视图上的工具条按钮提供了上下文菜单中一些项的快捷键。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

5.4.2 查看和修改变量和字段的值

一旦程序在断点处挂起并且已步进过它，则程序的流程可能不正确。例如，希望执行的 if 语句没有执行，或一些操作的期望输出不正确。此时还远远不能决定错误的原因。所能做的只是检查一些变量并判断其值是否正确。

有几种方法检查挂起程序中的变量值。对于简单变量，如果只需看一下它们的值的话，只需在 Java 编辑器视图中将鼠标覆盖在变量名上（不需单击）即可，这样将打开一个文本框并显示该变量的信息，包括它的当前值，如图 5.11 所示。



图 5.11 光标下的简单变量

这种方法对于简单变量和字符串变量很好用。对于复杂变量，只显示对象 ID 和它的类型，虽然这些信息也有用，但没有提供变量的值。如果光标下有与类型相关的任何 Javadoc 注释，同样也显示出来，如图 5.12 所示。

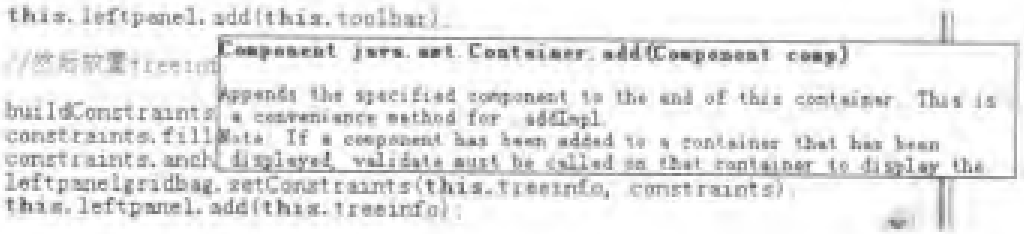


图 5.12 光标下的复杂变量

要显示复杂变量或希望在移走光标后显示变量的值，使用变量视图。默认情况下，变量视图位于窗口的右上角，如图 5.13 所示。

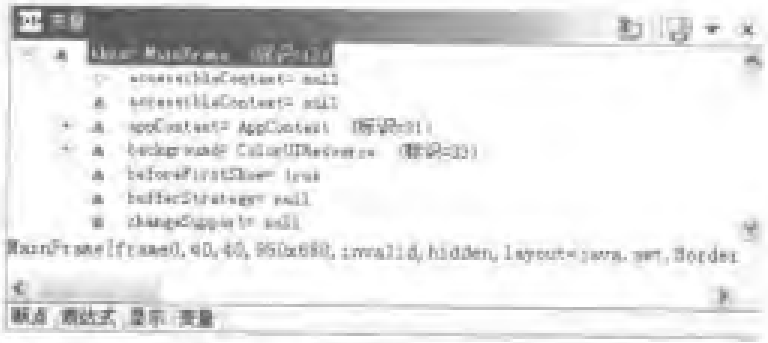


图 5.13 变量视图

在变量视图上还可以修改变量的值。选择变量，然后从它的上下文菜单，选择“修改变量的值”。而对于复杂变量，则需要选择变量的一个特定元素。变量视图显示当前在范围内的所有变量。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在变量视图中，原子类型和字符串类型直接显示它们的值。复杂变量可能包含其他一些变量，可以单击它们名字左边的加号（+）来展开。单击加号进一步展开类型，并可以下钻查看单个成分。

要检查一个变量，也可在 Java 编辑器中选择它的名字，并从其上下文菜单中选择“检查”，如图 5.14 所示。也可在变量视图中执行以下这些相同的步骤。



图 5.14　检查变量的上下文菜单

变量视图有几个工具条按钮，如图 5.15 所示，允许修改显示的信息。这些工具条按钮功能还可通过视图的上下文菜单获得。



图 5.15　变量视图工具条按钮

选择“显示限定名”按钮，则变量的类型以全称显示。如果没有选择这个按钮，则变量的类型不以全称显示。显示“类型名”按钮使得在变量名前显示变量的类型。“显示详细信息窗格”（Show Detail Pane）按钮则给出一个分屏视图。在上面窗格中选择的变量将在下面的窗格中显示它的值。

还有两个按钮是过滤器。它们控制在变量视图上是否显示静态变量和常量。注意，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

如不选择这些过滤器，则视图中就不会显示类型的成员。

5.4.3　条件断点

有些情况下，不希望每次遇到断点时都将程序挂起，这时可在行断点基础上设置条件。首先按前面讲述的方法设置一个断点。然后可在断点视图中的断点的上下文菜单中选择“属性”；或在 Java 编辑器的标记条中从断点本身的上下文菜单中选择“断点属性”，则打开“Java 行断点属性窗口”，如图 5.16 所示。

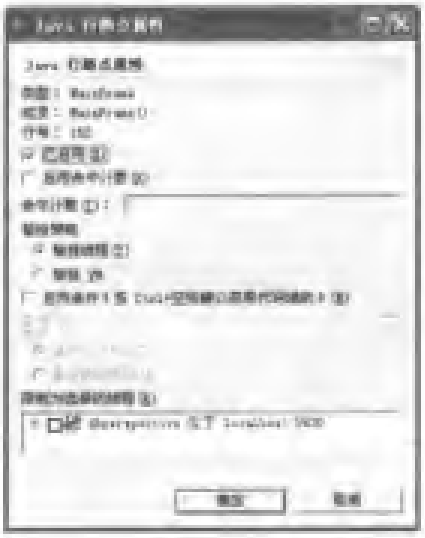


图 5.16　Java 行断点属性

在断点的属性中可选择“启用命中计数”或“启用条件”，分别对应了两类条件断点“命中计数”断点和“逻辑条件”断点。

启用命中计数则可在“命中计数”域中输入一个数字。这个数字表示该语句执行多少次后调试器会挂起程序。对于想要了解执行一定次数的循环后的情况，使用这种方法尤为有用。

达到命中计数阈值后，就挂起执行，并且禁用断点。挂起策略可分为两类：“挂起线程”和“挂起虚拟机”。前者挂起单个线程，后者将挂起整个 Java 虚拟机。

基于逻辑的条件断点在满足特定的条件时挂起执行。若在 Java 行断点属性选择“启用条件”，则可指定挂起条件。

可同时启用命中计数和条件。这将具有更大的灵活性。此外，条件断点在标记条上的蓝色球形断点图标上叠加了一个问号。

5.4.4　Java 异常断点

程序在执行时有时会抛出一个 Java 异常。如果没有捕获异常，程序就会异常终止，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

并且既不提供错误报告也不会试图恢复。对于这种情况，一个好的处理方法就是设置 Java 异常断点。

如要设置 Java 例外断点，可在断点视图中单击“添加 Java 异常断点”按钮，将打开“添加 Java 异常断点”窗口，如图 5.17 所示。



图 5.17 添加 Java 异常断点

在“添加 Java 异常断点”窗口中可选择一个 Java 异常并设置断点。窗口的底部使用复选框指示在发生异常处如何暂停执行：

- 如果希望在抛出并捕获异常处挂起执行，则选择“已捕获”；
- 如果希望在异常没有被捕获的地方挂起执行，则选择“取消捕获”。

在编辑器视图的标记条上不指示 Java 异常断点，因为这种类型的断点不与特定的代码行相关。在断点视图中可列出、修改或删除 Java 异常断点。

5.4.5 监视点

Eclipse 工作台提供了一个非常强大的调试器功能——监视点（Watchpoint），允许在访问或修改变量时挂起应用程序的执行。这对检测未经许可的或未知的变量访问尤为有用。可能某个线程或方法调用正在修改值的变量，这时访问的值就会不正确。

要设置监视点，在变量或表达式视图中选择变量并从其上下文菜单中选择“添加/除去监视点”，如图 5.18 所示。注意在选择这个选项前必须挂起程序的执行，并且所述的变量必须已显示在视图中。

监视点只在断点视图中列出。如图 5.19 所示，在断点视图中的监视点的上下文菜单中可使用以下操作：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 命中计数
- 修改
- 访问
- 启用
- 禁用

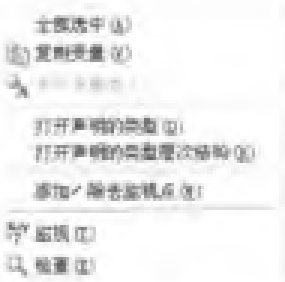


图 5.18　添加 / 除去监视点



图 5.19　监视点的操作

5.4.6　方法断点

方法断点只适用于二进制文件，在 Java 应用程序中它们可使用于类文件。如果类文件可以使用对应的源代码文件，则一旦遇到断点就可以步进代码进行常规调试。

当进入或退出方法调用时，方法断点将挂起执行。进入方法断点，在执行类中的任何代码之前挂起执行；退出方法断点在方法的结束括号处挂起。

要应用方法断点，首先打开一个类文件。在它的大纲视图中，选择希望添加断点的方法，从它的上下文菜单选择“添加 / 除去方法断点”，如图 5.20 所示。

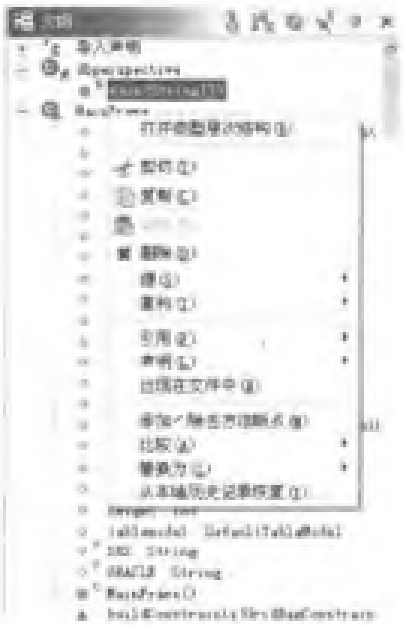


图 5.20　设置方法断点

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

方法断点只在断点视图中列出。在断点视图中的方法断点的上下文菜单中可使用以下操作：

- 命中计数
- 进入
- 退出
- 启用
- 禁用

5.4.7 代码片段编辑测试窗页面

代码片段编辑测试窗页面并不是调试器的一部分。将代码正式写进 Java 类之前，可使用代码片段编辑测试窗页面测试新代码并查看结果。

代码片段编辑测试窗页面的功能与调试器中的显示视图非常类似。可输入代码片段并分别运行。在代码片段编辑测试窗页面的上下文中可选择运行单行或多行代码。如选择多行，则必须是连续的。

在 Java 工程中创建代码片段编辑测试窗页面，然后在页中输入、选择并执行代码，使用 System.out.println()或类似的方法在控制台上显示结果。此外，还可计算表达式并显示或检查其结果。

在 Java 透视图中创建代码片段编辑测试窗页面页时，首先选择将包含代码片段编辑测试窗页面页的工程，在工程的上下文菜单中，选择“新建”→“代码片段编辑测试窗页面”，或在主菜单中选择“文件”→“新建”→“代码片段编辑测试窗页面”，如图 5.21 所示。第三种方法是在 Java 透视图上单击“创建代码片段编辑测试窗页面”工具条。

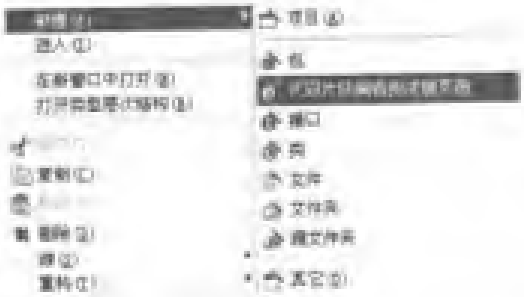


图 5.21 创建一个新的代码片段编辑测试窗页面

打开“创建 Java 代码片段编辑测试窗页面”窗口，如图 5.22 所示，可以命名页。任何遵循 Java 命名规范的名字都可使用。

向导在提供的名字后添加扩展名.jpage。下一步，在编辑器中打开代码片段编辑测试窗页面。

要做的第一件事通常是设置导入声明，这样就无须在代码片段中输入类型的完整名称。导入可在类或包级设置。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 5.22 命名 Java 代码片段编辑测试窗页面

打开 Java 片段导入对话框有多种方式。可从代码片段编辑测试窗页面的编辑器的上下文菜单选择“设置导入”，如图 5.23 所示。也可在 Java 透视图上单击“设置用于运行代码的导入声明”工具条按钮。无论采用何种方式，所打开的“Java 片断导入”窗口是一样的，如图 5.24 所示。



图 5.23 设置导入



图 5.24 Java 片段导入对话框

在“Java 片段导入”窗口中单击“添加类型”后，就打开“将类型作为导入来添加”窗口。在这个对话框，选择需要的类型并单击“确定”进行添加。注意在这个对话框，可以输入类型的前几个字母以缩小显示的类型列表。不要在输入域中输入包名；只需要实际类型的名称。如果选择的名字在多个包中存在，在单击“确定”按钮前还需要选择

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

对应的包。在选择了所有的类型或包之后，单击“确定”按钮返回代码片段编辑测试窗页面编辑器。

在代码片段编辑测试窗页面可输入任何 Java 代码。选中代码并从上下文菜单中，选择“运行”。在控制台视图中将显示运行的结果。

除了运行选择的片段，还可以检查和显示表达式。这些与调试器中的操作方式非常相似。检查表达式导致打开一个表达式视图，并在那里显示结果。表达式的返回值显示在表达式同一行的右边。

如果计算的表达式产生了运行时异常，在代码片段编辑测试窗页面编辑器中将会报告该异常。通常，上下文菜单中的操作也显示在 Java 透视图的工具条上，如图 5.25 所示。

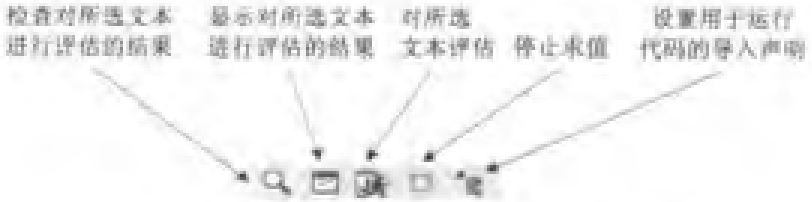


图 5.25　代码片段编辑测试窗页面工具条按钮

5.4.8　远程调试

开发者可以使用 Eclipse 调试在另一台机器上运行的 Java 程序，具体步骤如下。

① 启动 Eclipse。

② 切换到 Java 透视图。

③ 单击工具条上的“新建 Java 项目”按钮来创建一个新的 Java 项目。

④ 如果有程序的源文件，可将它导入到新项目中，这样调试器在调试时可以显示源文件。这也使得调试者可以在远程程序中设置断点。

⑤ 启动远程机器上的 Java 程序。确保在命令行参数上增加了以下行，从而可以调试 Java 程序：

```
-Xdebug -Xnoagent -
Xrunjdwp:transport=dt_socket,server=n,suspend=n,address=8000 -
Djava.compiler=NONE
```

调试者可以修改地址值指定所使用的端口，但需要记住修改值，以后会用到。

⑥ 选择刚才创建的 Java 项目。

⑦ 选择“运行”→“调试…”。

⑧ 选择远程 Java 应用，并单击“新建”。

⑨ 输入远程 Java 程序的主机和端口号，然后单击“完成”按钮。

⑩ 调试器将连接到远程 Java 程序，并切换到调试透视图。

这种设置强制调试者的 Java 程序在执行前等待调试器的连接。如果正在调试的程序不需在启动期间被调试，可以将命令行参数“server=n”改为“server=y”。这对调试 web 服务和类似的程序是有用的。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

5.5 热代码替换

如果使用调试器发现了代码中的问题，当然希望修改出错的代码或表达式。可以在调试透视图的 Java 编辑器视图中直接修改。修改代码后保存文件。整个功能称为热代码替换或热方法替换，这是 Java 虚拟机的功能，不需要重启 Java 虚拟机就可以在线更新载入的类。如果当前 Java 虚拟机不支持热代码替换，则显示错误消息提示。

出现警告的原因是在 Java 运行时环境（JRE）中运行程序的 Java 虚拟机不支持热代码替换。JVM 版本 1.4 及更高版本支持热代码替换。因此可以定义另一个 1.4 版本或更高版本的 JVM，在 Eclipse 工作台中使用该 JVM 运行并调试程序。这样做可以自动使调试器允许用户保存 Java 类并立即将修改的类载入到 JVM 中。在调试器中所做的任何修改无须重启应用程序就立即可用。

可从 www.javasoft.com 获取最新的 JRE。下载好 JRE 后，运行安装程序，在用户指定的目录中进行安装。

要在 Eclipse 中使用新的 JRE，选择“窗口”→“首选项”→“Java”→“已安装的 JRE”将打开首选项的“已安装的 Java 运行时环境”窗口，如图 5.26 所示。

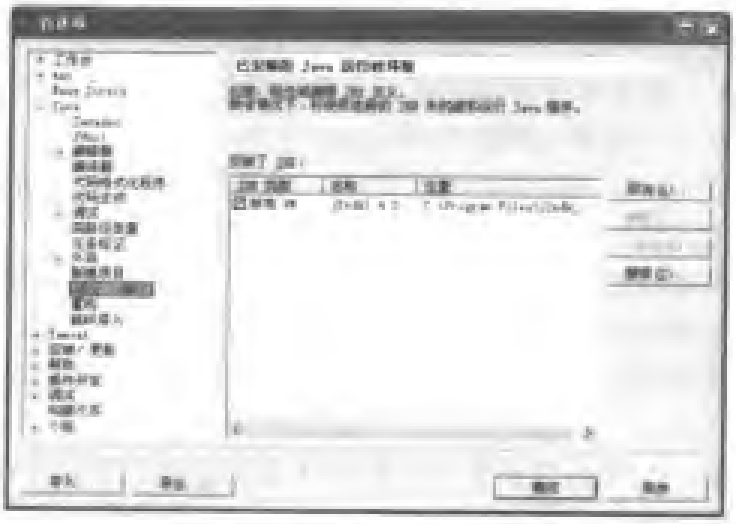


图 5.26 选择运行的 JRE

在首选项对话框的“已安装 Java 运行时环境”页上，单击“添加”。在添加 JRE 对话框中，将 JRE 主目录（home directory）域指向安装 JRE 的目录。单击“确定”添加 JRE。然后回到首选项页面，在刚安装的 JRE 边上的“JRE 类型”下选中复选框，然后单击“确定”。

现在可在调试器中修改代码并保存类。所做的修改在程序中将立即生效。如果运行的 JRE 类型为 1.4 版本或更高版本的 JVM，则不会出现警告信息。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第6章 JUnit

JUnit是用于Java的源代码单元测试框架。JUnit用于编写可重复测试的简单测试框架。JDT工具包括一个将JUnit集成到Java IDE中的插件。JUnit插件允许定义代码的回归测试并从Java集成开发环境中运行它们。本章将讨论JUnit的组件，以及如何在Eclipse环境中使用JUnit。有关JUnit的更多信息可参见http://www.junit.org/。

6.1 单元测试简介

软件测试是软件开发的重要组成部分之一。软件测试有不同的级别，如单元测试、集成测试、系统测试、用户测试（又称验收测试）和回归测试等。

JUnit测试框架是用于最低层次的Java应用程序测试——单元测试。单元测试是编码完成后，首先要实施的测试。单元测试检验单元编码与设计是否吻合，包括接口、方法、数据结构、边界、覆盖和逻辑等测试。主要由开发人员进行单元测试。

集成测试通常在主要的单元测试完成后进行，它通常与单元测试同步进行。集成测试主要用于测试组成系统的模块接口有无错误，以及检测代码实现的系统设计与需求定义是否吻合。主要由开发人员和测试人员协同进行集成测试。

系统测试通常在集成测试完成后进行。系统测试主要测试组成整个系统的代码、以及系统的软硬件有无错误；测试实现的系统与用户的需求是否吻合；检验系统的文档等是否完整、有效；模拟验收测试的要求，测试系统是否符合用户的验收标准。主要由测试人员进行系统测试。

系统测试完成后，在项目组看来开发和测试工作已经全部完成，可以交付使用时进行用户测试。用户测试检查系统是否符合实现约定的验收标准。由用户、开发人员、测试人员等一起协作完成用户测试。

回归测试在每次错误修改之后或版本更新之后进行。回归测试主要测试程序修改或版本更新之后，原先正确的功能是否仍然正确。回归测试主要由开发人员和测试人员协作完成。

6.1.1 单元测试概述

测试一直被视为项目开发结束时要做的事。当然，可以在项目进行之中结合测试，测试通常并不是在所有编码工作结束后才开始，而是一般在稍后阶段进行测试。JUnit是一个用来在项目中进行测试和调试的工具。使用JUnit可对软件开发中的测试活动进行组织和管理。然而，TDD（测试驱动开发）理论完全改变了这种开发方式。JUnit是实

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

现 TDD（测试驱动开发）理论的重要手段之一。

TDD 的主要理念是：在编写程序代码之前，必须首先编写好与之对应的自动测试。甚至程序代码并不存在，那也会看见一个失败的测试结果。作为一名程序员，应该在编写代码之前编写测试，然后只编写足以让测试通过的代码即可。对于每一个开发人员，只要有可能，就需要有更多的测试来检验代码。组织这些测试和组织程序代码一样重要。JUnit 就是用来让开发人员检验自己的 Java 代码或新编写的方法的软件。

测试驱动的开发是让单元测试（也称为程序员测试）决定开发者所编写的代码。这意味着开发者必须在编写代码之前进行测试。测试指出开发者需要编写的代码，从而也决定了开发者所要编写的代码。开发者只需编写足够通过测试的代码即可。

6.1.2 JUnit 测试的作用

软件测试是为了发现程序的错误而执行程序的过程。软件测试是保证软件质量的重要手段。虽然JUnit需要开发者花费相当多的时间为代码编写测试案例或测试套件，但JUnit能很有效地帮助开发者找出代码中的错误，并尽可能地为修正错误提供更多的信息。

JUnit 是静态测试技术。所谓静态测试是指，测试活动不实际运行程序，而是通过检查手段来发现错误并评估代码的质量。与静态测试技术对应的是动态测试技术。动态测试实际运行程序，并通过观测程序运行的实际结果来发现程序中存在的错误。

JUnit 是白盒测试。所谓白盒测试是指知道程序内部结构的情况下采用的测试技术或策略。与白盒测试相对应的是黑盒测试。黑盒测试是指，在不知道程序内部结构，只知道程序规格的情况下采用的测试技术或策略。

JUnit 从程序的最小单元开始检查程序中的错误，从而能够帮助开发者尽可能多、尽可能早地发现程序中的错误，而且测试的过程和数据可以复用。

JUnit 能够证明所开发的软件有错误，但不能证明软件没有错误。其实在软件开发的实际项目中，尚未有任何一种软件工具能帮助证明所开发的软件没有错误。

6.1.3 JUnit 的获取和安装

JUnit 是由 Erich Gamma 和 Kent Beck 编写的开放源代码软件。在下列 URL 可以选择下载的镜像站点，并下载 JUnit：

http://prdownloads.sourceforge.net/junit/junit3.8.1.zip?download

目前，JUnit 的版本是 3.8.1。JUnit 的安装非常简单，其主要步骤如下：

① 将 junit-3.8.1.zip 解压缩到任一目录下，此目录即为 JUnit 的安装目录；

② 将 junit.jar 和安装目录下的\junit 子目录添加到类路径中。

Eclipse 中包含了 JUnit 组件，无须另行下载和安装 JUnit。TestRunner 是运行 JUnit 测试案例和 JUnit 测试套件的用户界面。可以在 TestRunner 中独立运行单元测试。然而，Eclipse 也提供了运行 JUnit 单元测试案例和 JUnit 测试套件的图形用户界面，可以在 Eclipse 中运行 JUnit 单元测试。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

6.1.4 Eclipse 中 JUnit 的设置

在 Eclipse 首选项中可以设置 JUnit，设置步骤如下。

① 在 Eclipse 的工作台主菜单中选择“窗口”→“首选项”，打开首选项设置页面。

② 在首选项设置页面的左边窗格展开“Java”类别，然后选中“JUnit”条目，打开 JUnit 设置页面，如图 6.1 所示。



图 6.1 JUnit 设置

在图 6.1 中，“仅当发生错误或故障时才显示 JUnit 结果视图”复选框表示仅当发生了错误或故障时才会显示 JUnit 视图，该复选项的默认值为启用。堆栈跟踪过滤器模式表示了不应显示在测试故障的堆栈跟踪中的包、类或模式。“堆栈跟踪过滤器模式”列表框中显示了默认的过滤器模式，可以添加或除去过滤模式。

6.2 JUnit 测试案例

使用 JUnit 可以对所编写的测试案例进行单元测试。测试案例对应为 Java 类，并包含对类的方法进行测试的代码。这些测试案例是作为 TestCase 的子类来实现的，而 TestCase 是 JUnit 框架的一部分。因此，可对测试案例进行初始化，并在测试完成后清除它们。测试案例的初始化是通过重载 setUp 方法实现，而测试案例的清除是通过重载 tearDown 方法实现的。

6.2.1 在 Eclipse 中创建测试案例

将测试类和所测试的类分开是一种比较好的处理方式，所以通常应创建一个独立的包，用来放置测试类。需要注意的是，需要将 JUnit 的 jar 文件添加到项目的构建路径中，否则在项目中将找不到 JUnit 类。如图 6.2 所示，junit.jar 文件位于 Eclipse 安装目录下的\plugins\org.junit_3.8.1 子目录中。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

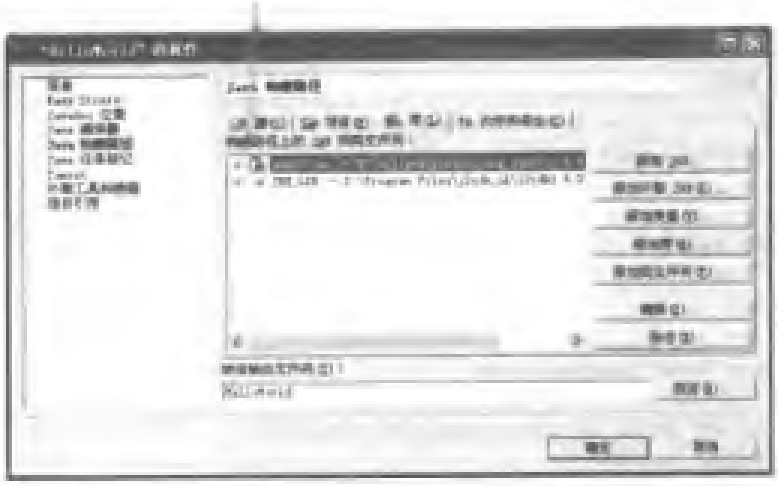


图 6.2　在项目的构建路径中添加 junit.jar 文件

创建好用于存放测试案例的包后，即可创建测试案例。Eclipse 提供了创建测试案例的向导，具体步骤如下。

① 在包资源管理器视图中选中测试包。

② 在上下文菜单中选择“新建”→“其他”，此时将打开“新建”窗口。

③ 在“新建”窗口的左侧窗格展开“Java”类别，并选中它下面的“JUnit”条目；然后在右侧窗格中选择“测试案例”，如图 6.3 所示。

④ 单击“下一步”按钮，将打开“新建 JUnit 测试案例”窗口，如图 6.4 所示。



图 6.3　新建 JUnit 测试案例



图 6.4　新建 JUnit 测试案例

⑤ 在“测试案例”字段中输入测试案例名，此例中假设是“FirstTestCase”。

⑥ 选中“public static void main(String[] args)”复选框、“setup()”复选框和“teardown()”复选框。

⑦ 单击“完成”按钮，将在测试包中自动创建一个测试案例类 FirstTestCase.java。在编辑器中打开 FirstTestCase.java，其代码如下所示：

```
package testPackage;
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
import junit.framework.TestCase;

public class FirstTestCase extends TestCase {
    public FirstTestCase(String arg0) {
        super(arg0);
    }

    public static void main(String[] args) {
    }

    /*
     * @see TestCase#setUp()
     */
    protected void setUp() throws Exception {
        super.setUp();
    }

    /*
     * @see TestCase#tearDown()
     */
    protected void tearDown() throws Exception {
        super.tearDown();
    }

}
```

开发者可以在 FirstTestCase.java 中编写 setUp()和 tearDown()方法，从而在运行测试之前初始化资源，并在测试完成后清除资源。需要注意的是，开发者并不一定需要编写 setUp()或 tearDown()方法；只有当测试方法共享某些公共的设置或销毁活动时，才需要创建它们。

6.2.2 向测试案例中添加测试方法

在 TestCase 类中的任何名字以 test 开头的方法都是测试方法。测试方法必须是 public，并且返回 void。

开发者可以编写多个测试方法。每一个测试方法都将用到多个断言（assert）方法。断言方法用于实际测试所测试的类的状态。断言方法测试事情是否为“真（true）”。换句话说，这些方法针对某件事做出一个断言，该断言告诉 JUnit 验证该事是否为真。断言方法从测试案例中继承。断言方法包括：

- assertEqual (x,y)

比较两个基本类型或对象是否相等。

- assertFalse (boolean)

对布尔值求值，看它是否为“真”。

- assertTrue (boolean)

对布尔值求值，看它是否为“假（false）”。

- assertNull (object)

检查对象是否为“空（null）”。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- assertNotNull (object)

检查对象是否不为“空”。

- assetSame (firstObject, secondObject)

检查两个对象是否为同一实例。

- assertNotSame (firstObject, secondObject)

检查两个对象是否不为同一实例。

假设我们使用 assertEqual (x,y)断言方法，向测试案例 FirstTestCase.java 中添加测试方法。测试方法的代码如下：

```
public void testCompareSucceed(){
        assertEquals(0,0);  //本断言会成功
    }

public void testCompareFail(){
        assertEquals(0,1); //本断言将失败
    }
```

6.2.3 运行测试案例

在 Eclipse 工作台中，可直接运行测试案例类，其步骤如下。

① 选中测试案例类。

② 在 Eclipse 工作台主菜单中选择“运行”→“运行为”→“JUnit 测试”，如图 6.5 所示。



图 6.5 运行测试案例

此时，Eclipse 将运行测试案例中的各类方法。在执行每个测试方法之前执行 setUp() 方法，接着调用测试方法中正在测试的方法，在执行每个测试方法之后执行 tearDown() 方法。同时，Eclipse 将自动在工作台窗口中打开 JUnit 视图，如图 6.6 所示。

在 JUnit 视图中，将会显示测试的执行情况。在图 6.6 中，JUnit 视图的顶部显示了一个红条，这表明测试方法中出现了失败。如果没有出现失败，则在出现红条的地方会显示一个绿条，这表示测试方法全部成功。

在红条下面，“运行次数”显示了共有几个测试方法，本次运行执行了几个方法；“错误次数”表示了当抛出异常时，出现了几个错误；“故障次数”显示共有几个测试方法失败了。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 6.6 JUnit 视图

在红条下的“故障次数”窗格中将显示哪一个测试导致了失败。在图 6.6 中，可以看到 testCompareFail()方法失败了。

如果希望查看测试案例中的测试方法，可以单击“故障次数”选项卡旁的“层次结构”选项卡。这将会显示测试案例中的所有测试方法，如图 6.7 所示。在层次结构中的所有测试方法的最左边将有一个图标显示测试“成功”还是“失败”，“√”表示成功；“×”表示失败。

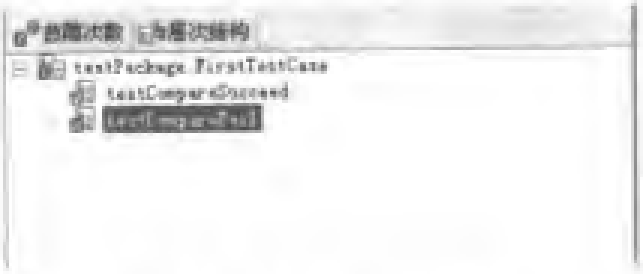


图 6.7 测试方法的层次结构

在“故障跟踪”窗格中，将可以看到导致失败的调用跟踪，从而开发者可以查看失败的详细信息。

现在，我们修改测试案例 FirstTestCase.java 中的测试方法，将测试方法 testCompaeFail()的代码改为：

```
public void testCompareFail(){
        assertEquals(1,1);
}
```

保存修改，并重新运行测试案例，可看到 JUnit 视图中状态条的颜色由红色变为绿色，这表示测试案例中的所有测试方法全部执行成功，如图 6.8 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

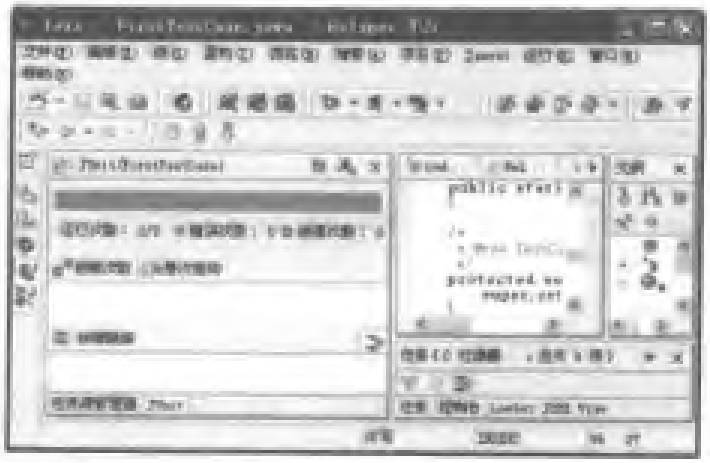


图 6.8　成功执行测试案例中的所有测试方法

6.2.4　创建第二个测试案例

此时，假设在测试包中将创建另一个测试案例 SecondTestCase.java，该测试案例将测试项目中类 HelloWorld.java 中的方法，其代码为：

```
package cn.edu.sjtu.hello;

public class HelloWorld {

    public static void main(String[] args) {
        System.out.println(say());
    }

    public static String say() {
        return ("Hello World!");
    }

}
```

创建本测试案例的步骤与前面所述的创建第一个测试案例的步骤一样，但所添加的测试方法不再是前面带有演示目的的 testCompareSucceed()和 testCompareFail()，而是叫做 testSay()的方法。这个方法调用了 assertEquals()断言方法，它用来比较预期的值"Hello World!"和由 HelloWorld 类中的 say()返回的值。

第二个测试案例 SecondTestCase.java 的代码如下：

```
package testPackage;

import junit.framework.TestCase;
import cn.edu.sjtu.hello.*;

public class SecondTestCase extends TestCase {

    public SecondTestCase(String arg0) {
        super(arg0);
    }
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    public static void main(String[] args) {
    }

    protected void setUp() throws Exception {
        super.setUp();
    }

    protected void tearDown() throws Exception {
        super.tearDown();
    }

    public void testSay() {
        HelloWorld hi = new HelloWorld();
        assertEquals("Hello World!", HelloWorld.say());
    }

}
```

在 Eclipse 工作台中运行测试案例 SecondTestCase.java，在 JUnit 视图中可以看到该测试案例运行正确，这表明 Hello 类中的 say()方法确实返回了我们所期望的值“Hello World!”。

6.2.5 编写测试案例的考虑事项

一个不好的测试案例往往既浪费了时间，也起不了实际的作用。相反，一个好的测试案例，不仅可以很好地指出代码中存在的问题，而且也可以作为代码更准确的文档，同时还在系统集成的过程中起非常重要的作用。下面是编写测试案例时需要注意的地方。

- 会失败的测试是最重要的测试

在 JUnit 的测试中，真正有用的是那些会失败的测试，即那些开发者认为不会失败的测试；或者是那些开发者认为应该失败，却成功的测试。

- 测试是一个不间断的过程

在 JUnit 中，测试是一个不间断的过程。一旦有一个测试，就要一直确保其正常工作，以检验所编写的新增代码。

- 测试的独立性

一次只测试一个对象，方便定位出错的位置。这有两层意思：一层含义是一个测试案例只测试一个对象；另一层含义是一个测试方法只测试这个对象中的一个方法。

- 给测试方法一个合适的名字

测试类和测试方法应该有一个统一的命名方案。在断言方法中给出失败的原因，从而方便查错。

- 测试所有可能引起失败的地方

对于 Eclipse 产生的代码可以不进行测试；对于开发者手工编写的代码，最好全部进行测试。

- setUp()和 tearDown()中的代码应该是全局相关的

在 setUp()方法和 tearDown()方法中的代码不应该是与测试方法相关的，而应该是全局相关的。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

6.3 测试套件

测试套件可运行多个测试案例或测试套件。和测试案例一样，测试套件也是 JUnit 测试框架中的类，它可作为 TestSuite 的子类实现。测试套件可将多个测试案例组合起来，并一起运行。

6.3.1 在 Eclipse 中创建 JUnit 测试套件

在 Eclipse 中创建 Eclipse 测试套件的方式和创建测试案例类似，都可通过向导创建，其步骤如下。

① 在包资源管理器视图中选中测试包。

② 在上下文菜单中选择“新建”→“其他”，此时将打开“新建”窗口。

③ 在“新建”窗口的左侧窗格展开“Java”类别，并选中它下面的“JUnit”条目；然后在右侧窗格中选择“测试套件”，如图 6.9 所示。

④ 单击“下一步”按钮，将打开“新建 JUnit 测试套件”窗口，如图 6.10 所示。



图 6.9 新建 JUnit 测试案例



图 6.10 新建 JUnit 测试套件

⑤ 在“测试套件”字段中输入测试套件名，此例中假设是“FirstTestSuite”。

⑥ 选择需要添加到测试套件中的测试案例类，在本例中将选择前面创建的两个测试案例类 FirstTestCase 和 SecondTestCase。

⑦ 单击“完成”按钮，将在测试包中自动创建一个测试套件类 FirstTestSuite.java。在编辑器中打开 FirstTestCase.java，其代码如下所示：

```
package testPackage;

import junit.framework.Test;
import junit.framework.TestSuite;

public class AllTests {
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    public static Test suite() {
        TestSuite suite = new TestSuite("Test for testPackage");
        //$JUnit-BEGIN$
        suite.addTest(new TestSuite(FirstTestCase.class));
        suite.addTest(new TestSuite(SecondTestCase.class));
        //$JUnit-END$
        return suite;
    }
}
```

6.3.2 运行测试套件

在 Eclipse 工作台中运行测试套件的步骤，与运行测试案例一样，只需选中相应的测试套件类，然后在 Eclipse 工作台按钮中选择“运行”→“运行为”→“Junit”测试。Eclipse 将运行测试套件类，并将运行结果显示在 JUnit 视图中。与运行单个测试案例不同的是，运行测试套件将会执行测试套件中所包含的所有测试案例。

测试套件 FirstTestSuite.java 的运行结果在 JUnit 视图的“层次结构”窗格中的显示如图 6.11 所示。



图 6.11 测试套件 FirstTestSuite.java 的层次结构

从图 6.11 中可以看出，测试套件 FirstTestSuite.java 的运行共包含两个测试案例中的三个方法。

6.4 测试案例和测试套件的调试执行方式

测试案例和测试套件的执行方式除了如前面所述的可以在“运行”方式下启动，还可以在“调试”方式下启动。在“调试”方式下启动测试案例或测试套件的途径有两种。

第一种方式，首先选择相应的测试案例或测试套件，然后在 Eclipse 的 Java 透视图中单击 Eclipse 工作台中的“调试”按钮，再依次选择“测试为”→“JUnit 测试”，如图 6.12 所示。

第二种方式，首先选择相应的测试案例或测试套件，然后在 Eclipse 的 Java 透视图中，选择主菜单中的“运行”→“调试为”→“JUnit 测试”，如图 6.13 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 6.12　使用调试按钮执行 JUnit 测试



图 6.13　使用菜单执行 JUnit 测试

无论采用哪种方式，都将打开调试透视图。在调试透视图中将执行相应的测试案例或测试套件，并将执行结果显示在打开的 JUnit 视图中，如图 6.14 所示。

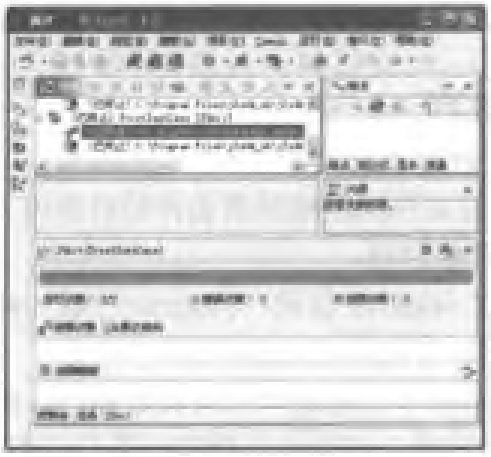


图 6.14　在调试模式下执行测试案例或测试套件

从测试案例和测试套件的代码可知：JUnit 类，无论是测试案例还是测试套件，它们都实现了 JUnit 的 Test 接口。因此，开发者可以将 JUnit 测试套件添加到其他的测试套件中，从而形成测试套件的嵌套结构，从而可以同时运行多个测试套件。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 7 章　Eclipse 配置管理

启动工作台时，用户并未启动单个 Java 程序。在激活平台运行时的时候，它可以动态地发现已注册的插件并根据需要来启动它们。在平台的透视图中，用户开发的插件与基本插件没有任何不同。

7.1　Eclipse 的结构

7.1.1　Eclipse 目录内容

安装 Eclipse SDK 时，将文件解压缩到所选择的目录中。该目录称为 Eclipse 的安装目录，也称为 Eclipse 平台的根目录。

Eclipse 安装目录中包含 features 目录、plugins 目录、workspace 目录及其他目录。在 Eclipse 3.0 的安装目录中，还包括 configuration 子目录。

workspace 目录用来保存由平台创建和处理的资源。在 workspace 目录中包含工作区中存在的每个项目的不同子目录，这些子目录称为项目的内容目录。内容目录是用户创建项目时由平台创建的。内容目录中除了有一个.project 文件，其文件和文件夹的布局与它们在工作区的资源树中的布局完全相同。每个项目有其自己的.project 文件，在其中保留关于项目的元数据。

workspace 中还包含一个.metadata 目录。有关工作区结构的重要信息，如项目的引用或者资源的属性等，都存储在该目录中。

除了.metadata 目录和.project 文件之外，其他工具，如文本编辑器和文件系统实用程序，也可使用和处理工作区目录中的文件夹和文件。工作台提供了刷新操作来协调资源的工作区视图与文件系统的实际状态。

7.1.2　功能部件的作用

基于 Eclipse 的大型产品可以将它们的功能部件组织成从表示整个产品的根功能部件开始的树。此根功能部件包括一些较小的功能单元，直至列示一个或多个插件和段的叶功能部件。以层次结构方式对功能部件进行分组的功能使得大型产品可在较小型产品的基础上进行构建，即通过在较小型产品添加更多的功能部件来实现大型产品。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

7.1.3 标识运行时平台中的功能部件

一些被包括的功能部件可能是很有用的添加件，但是它们对于整个产品的正确运行并不是十分重要的。功能部件提供程序可以选择将这些功能部件标记为可选的。当安装可选的功能部件时，用户可选择是否要安装它们。如果不立即安装可选功能部件，可在以后再添加它们。

帮助菜单上的“关于”选项提供了有关已安装功能部件和插件的信息。帮助菜单上的“软件更新”子菜单将各项集中在一起以更新现有功能部件以及查找、下载和安装新的功能部件。

7.2 安装 / 更新透视图和相关视图

在 Eclipse 中，可以使用更新管理器安装新的插件。Eclipse 更新管理器可用来从基于 Web 的特殊 Eclipse 更新站点中发现、下载和安装已更新的功能部件和插件。在 Eclipse 工作台主菜单中选择“帮助”→“软件更新”→“更新管理器”或选择“窗口”→“打开透视图”→“安装 / 更新”可打开安装 / 更新透视图。如图 7.1 所示，安装 / 更新透视图主要包含以下几个相关视图：

- 安装配置视图
- 功能部件更新视图
- 预览视图



图 7.1 安装 / 更新透视图

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 外部预览视图
- 所选更新视图
- 功能部件搜索结果视图

7.2.1 安装配置视图

通过“安装配置”视图，开发者可管理工作区配置。如图 7.2 所示，“Eclipse 平台”当前配置文件夹显示哪些本地安装位置是工作区配置的一部分，以及在哪些位置中安装了哪些功能部件；“配置历史记录”文件夹显示对工作区配置所作更改的历史记录；“已保存配置”文件夹显示已保存的所有配置。

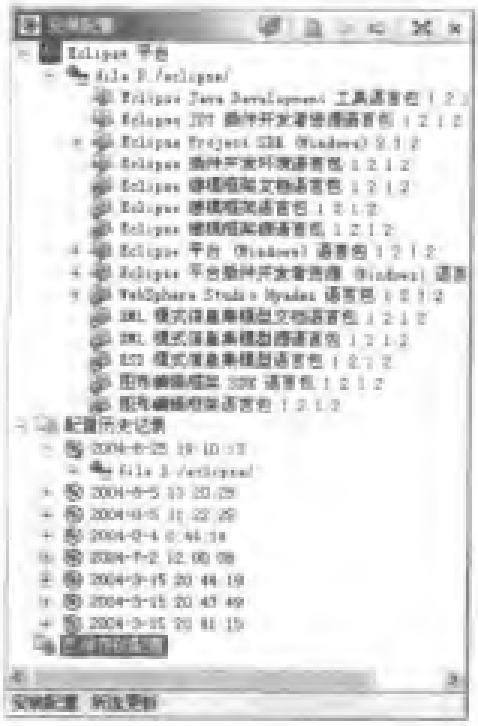


图 7.2 安装配置视图

与工作台中的其他视图一样，“安装配置”视图有它自己的工具栏。提供工具栏按钮是为了浏览以及切换视图以显示当前禁用的功能部件。如图 7.2 所示，安装配置视图的工具栏按钮包括：

- 显示禁用的功能部件

此命令切换视图以显示已安装但当前禁用的功能部件。

- 主页

此命令重新刷新视图并显示标准配置文件夹。

- 后退

此命令重新刷新视图并显示在当前显示内容之前所显示的那个层次结构。

- 进入

此命令重新刷新视图并使所选项的视图层次结构重新成为父代。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

7.2.2 功能部件更新视图

使用“功能部件更新”视图可以定位和安装新功能部件，以及更新功能部件。该视图包含三个标准文件夹，“要访问的站点”、“可用的更新”和“我的电脑”，如图 7.3 所示。



图 7.3 功能部件更新视图

“要访问的站点”文件夹包含正在使用的产品预先配置的更新站点。“可用的更新”是用于搜索更新站点的内置搜索结果文件夹。“我的电脑”文件夹是一个文件资源管理器，可用来导航至位于文件系统中的更新。

除标准文件夹之外，可通过添加更新站点书签，在“功能部件更新”视图中创建附加内容和定制搜索。

有许多适用于 Eclipse 平台的插件。在 eclipse.org 上，Eclipse 工具项目和 Eclipse 技术项目提供插件。Eclipse 工具项目，为 Eclipse 平台开发大量各种高质量的工具插件。用户可以访问 Eclipse 工具项目下载页来查看有哪些插件，并了解其他各个子项目。在 eclipse.org 上的 Eclipse 技术项目，提交在教育、研究和孵化器活动中开发的插件。技术项目的下载页也是插件的主要资源之一。此外，Eclipsep 插件开发者社区开始为 Eclipse 平台构建大量各种各样的插件。Eclipse.org 在社区插件页上维护这些插件的列表。这些项目中的许多都放在 www.sourceforge.org 上。

与“安装配置”视图一样，“功能部件更新”视图有它自己的工具栏。在“功能部件更新”视图中除了有和“安装配置”视图类似的工具按钮，还有“刷新”工具按钮和下拉菜单，如图 7.3 所示。单击“刷新”按钮可以刷新视图的内容。标题栏中的下拉菜单可切换下列设置：

- 显示我的电脑文件

切换“我的电脑”资源管理器是只显示目录结构还是显示所有文件。

- 显示站点类别

切换“更新站点”上的功能部件是显示在它们各自的类别中还是作为平面列表显示。

- 只显示匹配的功能部件

切换“更新站点”是只显示与当前执行环境相匹配的功能部件还是显示所有功能部件。

7.2.3 预览视图

“预览”视图显示有关当前活动选择的详细信息，如图 7.4 所示。当选择更改时，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

其内容也跟着更改。在某些情况下，“预览”只是选择的扩展描述，或者是根据选择提供的建议操作的页面。在其他情况下，可以使用预览来针对选择执行特定的操作。当前配置的预览显示有关为工作区定义的基本功能部件的信息。



图 7.4　预览视图

所选功能部件的预览显示有关该功能部件的相应安装信息。如果可以针对该选择执行某个操作，则在预览中显示为操作按钮。

所选历史记录的预览显示有关历史记录条目的相应配置信息。如果可以针对该选择执行某个操作，则在预览中显示为操作按钮。

类似地，“预览”视图具有相应的工具栏。

- 转至“更新管理器”首页

此命令按钮将显示“欢迎使用 Eclipse 更新”页面。

- 后退

此命令回到“预览”页面链中的上一页。

- 前进

此命令前进至“预览”页面链中的下一页。

7.2.4　外部预览视图

“外部预览”视图是用来显示 HTML 信息的嵌入式浏览器，如图 7.5 所示。此视图仅在 Windows 系统上可用。一般情况下，访问 HTML 页面时才会使用“外部预览”。

“外部预览”视图的工具栏按钮包括：

- 后退

此命令回到“预览”页面链中的上一页。

- 前进

此命令前进至“预览”页面链中的下一页。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 7.5 外部预览视图

- 执行

此命令导航至指定的“外部预览”页面。

- 停止

此命令停止装入“外部预览”页面。

- 刷新

此命令重新装入“外部预览”页面。

7.3 更新管理

更新管理器的基本底层机制很简单：功能部件或插件的文件总是存储在其名称包括版本标识符的子目录中。总是对功能部件或插件的不同版本指定不同版本标识符，从而确保同一功能部件或插件的不同版本可以在磁盘上共存。这表示安装或更新功能部件和插件需要添加更多文件，但从不需要删除或覆盖现有文件。

在本地计算机上安装文件之后，就可以配置新的功能部件和插件版本。因此，同一已安装文件基本部分能够同时支持许多不同的配置；安装和升级现有产品减少为以增量式方式规划比当前配置要新的配置。还可以保存重要的配置并在升级不成功的情况下恢复至活动服务。

7.3.1 在 Eclipse 中安装插件

在 Eclipse 平台中，很容易将附加功能部件和插件安装到基于 Eclipse 的产品以及更新产品的现有功能部件和插件中。可通过传统的安装方式，即解压缩到 Eclipse 安装目录下的/plugins 子目录中。也可使用 Eclipse 平台本身的更新管理器实现此目的，其具体步骤如下。

① 可以通过单击“帮助→软件更新→更新管理器”来打开主更新管理器。这将打开“安装 / 更新”透视图。为发布 Eclipse 功能部件和插件所在的 Web 站点创建书签。在

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

“功能部件更新”视图中，从上下文菜单中选择“新建→站点书签”。

② 在“功能部件更新”视图中展开新创建的项，以显示在该更新站点中可用的功能部件版本的类别。这将访问 Web 站点以发现哪些功能部件可用。展开顶级类别之一，并选择一个功能部件。有关该功能部件的信息将出现在“预览”视图中。在选择了要添加的功能部件之后，单击“预览”视图中的安装按钮。

③ “功能部件安装”向导会确认将要安装的功能部件的详细信息。单击“下一步”按钮，仔细复查已升级功能部件的许可证协议。如果所有这些许可证的条款都是可接受的，则选择“我接受许可证协议中的条款”。如果许可证条款不可接受，则不要继续下载该功能部件。

④ “安装位置”页面控制要在本地计算机上的何处安装新功能部件的文件。选择要将产品安装到其中的目录，并单击下一步。(如果将产品安装在一个用户对其没有写访问权的目录中，则应与系统管理员联系，并让他们安装此功能部件，以便它在通常情况下可用。另一选项是单击“添加”按钮并指向用户具有写访问权的目录；但是，仅当使用当前工作区时，安装在该处的功能部件才可用)。

⑤ 功能部件版本可以由提供它们的公司以数字形式签名。这使用户能够更容易验证要下载并安装的功能部件和插件是否都来自可信赖的供应商。由于插件可能具有危害甚至是怀有恶意的，所以只应从用户信任的组织中下载功能部件。

⑥ 一旦成功地下载了新功能部件和插件并将它们的文件安装到本地计算机上的产品中，就将阐明组合了这些功能部件和插件的新配置。当询问是否退出时单击“是”，并重新启动工作台以使更改生效。要在重新启动之前同时添加其他新功能部件，可反复单击“否”。

7.3.2 在 Eclipse 中更新插件

可使用 Eclipse 平台本身的更新管理器更新已有的插件，其具体步骤如下。

① 在 Eclipse 工作台的主菜单中选择“帮助”“软件更新”“新的更新”。这将跟与产品的功能部件相关联的 Web 站点联系，以发现哪些功能部件的哪些版本可用。“可用的更新”页面中提供了潜在的升级。

② 选择想要升级的功能部件版本，并单击“下一步”按钮。

③ 仔细复查已升级的功能部件的许可证协议。如果所有这些许可证的条款都可以接受，则选择“我接受许可证协议中的条款”。如果许可证条款不可接受，则不要继续下载功能部件。

④ 功能部件版本可以由提供它们的公司以数字形式签名。这使用户能够更容易验证要下载并安装的功能部件和插件是否都来自可信赖的供应商。单击“安装”继续进行下载和安装。

⑤ 一旦成功地下载了所有功能部件和插件并将它们的文件安装到本地计算机上的产品中，就将阐明组合了这些功能部件和插件的新配置。当询问您是否退出时，单击“是”，并重新启动“工作台”以使更改生效。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

7.3.3 使用组更新一次安装几个功能部件

在 Eclipse 中如要安装多个功能部件，既可如前所述，逐个安装功能部件，也可使用更新管理器一次安装几个功能部件，其具体步骤如下。

① 在 Eclipse 工作台的主菜单中选择“帮助”→“软件更新”→“更新管理器”，这将打开“安装 / 更新”透视图。

② 在“功能部件更新”视图中，从上下文菜单中选择“新建”→“站点书签”。为发布 Eclipse 功能部件和插件所在的 Web 站点创建书签。

③ 在打开的“新建站点书签”对话框中为站点命名，并输入相应的 URL。

④ 在“功能部件更新”视图中展开新创建的项，以显示在该更新站点中可用的功能部件版本的类别。这将访问 Web 站点以发现哪些功能部件可用。

⑤ 展开顶级类别中的其中一个，并选择一个功能部件。有关该功能部件的信息将出现在“预览”视图中。

⑥ 在选择了要添加的功能部件之后，找到带有标题“组更新”的“预览”视图中的一节，并选择“通过添加至所选更新进行安装”复选框。

⑦ 对想要安装的每个功能部件重复上述步骤。可随时通过按复选框右边的“所选的更新”超链接或通过将“所选的更新”视图移动到顶部来检查该选择。

⑧ 当选择了要安装的所有功能部件时，切换至“所选的更新”视图，并从弹出菜单中选择“全部处理”项。如果对任何功能部件，用户放弃了安装的计划，可在处理之前从列表中除去它。

⑨ “功能部件安装”向导会确认将要安装的功能部件的详细信息。如果已选择的功能部件之间存在冲突，系统将通知用户有关冲突，并要求取消选择导致冲突的功能部件。然后单击“下一步”按钮。

⑩ 仔细复查已升级的功能部件的许可证协议。如果所有这些许可证的条款都是可接受的，则选择“我接受许可证协议中的条款”。如果许可证条款不可接受，则不要继续下载该功能部件。

⑪ 如果选择要安装的一个或多个功能部件包括可选功能部件，则将显示一页，允许选择是否想要安装它们。可选功能部件通常是对主功能部件的正常运行并非必需的功能部件。

⑫ “安装位置”页面控制要在本地计算机上的何处安装新功能部件的文件。选择产品的安装目录，并单击“下一步”按钮。

⑬ 功能部件版本可以由提供它们的公司以数字形式签名。这将使得用户能够更容易验证要下载并安装的功能部件和插件是否都来自可信赖的供应商。

⑭ 单击“安装”按钮以让下载和安装继续进行。

⑮ 一旦成功地下载了新功能部件和插件并将它们的文件安装到本地计算机上的产品中，就将声明组合了这些功能部件和插件的新配置。当询问是否退出时单击“是”，并重新启动工作台以使更改生效。要在重新启动之前同时添加其他新功能部件，可单击

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

“否”并重复这些操作。

7.4 Eclipse 中的插件配置管理

7.4.1 检查 Eclipse 平台的当前配置

在 Eclipse 工作台的主菜单中选择“帮助”→“关于 Eclipse 平台”，将打开“关于 Eclipse 平台”对话框窗口，如图 7.6 所示。

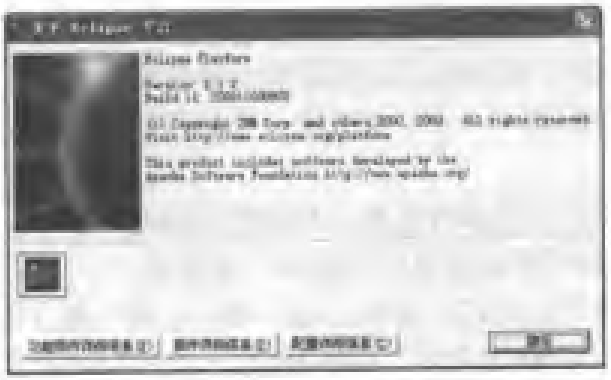


图 7.6 关于 Eclispe 平台

“关于 Eclispe 平台”窗口中将显示有关在产品中配置的功能部件和插件的大多数重要信息，包括功能部件详细信息、插件详细信息和配置详细信息。

在“关于 Eclispe 平台”窗口中单击“功能部件详细信息”按钮，将打开“功能部件”窗口，如图 7.7 所示。



图 7.7 关于 Eclipse 平台——功能部件

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在“功能部件”窗口中将显示在 Eclipse 平台中的所有功能部件。选中某一功能部件，并单击“详细插件信息”按钮，则将打开“功能部件插件”窗口，并在窗口中显示该功能部件向 Eclipse 平台所添加的插件。

在“关于 Eclipse 平台”窗口中单击“配置详细信息”按钮，则将打开显示 Eclipse 平台中的所有插件的窗口，如图 7.8 所示。



图 7.8 Eclipse 平台中的插件

在“关于 Eclipse 平台”窗口中单击“插件详细信息”按钮，则将在“安装 / 更新”透视图中打开“配置详细信息”视图，如图 7.9 所示。



图 7.9 “配置详细信息”视图

在 Eclipse 工作台中，执行以下步骤则可更详细地了解当前配置的有关信息。

① 在 Eclipse 工作台的主菜单中选择“帮助”→“软件更新”→“更新管理器”，这

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

将打开“安装 / 更新”透视图。

② 在“安装配置”视图中，展开“当前配置”项。这些项的第一层是本地计算机上的一些位置，功能部件和插件的文件就存储在这些位置中。对于典型产品，所有功能部件都安装在单个目录子树中。但是，作为另一产品的扩展安装的功能部件通常位于独立的目录子树中。

③ 展开该位置以查看安装在那里的功能部件版本。

④ 功能部件的嵌套指示哪些子功能部件包括在给定的父功能部件中。嵌套的功能部件充当一个单元。

⑤ 选择功能部件版本，则在“预览”视图中显示有关所选功能部件的更详细信息。

⑥ 单击“显示禁用的功能部件”工具按钮，则会显示已禁用的任何功能部件版本。禁用的功能部件版本就是安装在本地计算机上，但是从当前配置排除的版本。在给定配置中，最多可以存在任何给定功能部件的一个版本；通常具有较低版本号的版本将处于禁用状态。

⑦ 选择一个功能部件版本，并在上下文菜单中选择“显示状态”，以了解是否完全像封装那样配置了所选择的功能部件。状态将显示功能部件的插件何时存在于不同版本中，或者由于丢失了希望由其他一些功能部件提供的必备插件而被禁用。

7.4.2 保存 Eclipse 平台的当前配置

为了提高升级产品或安装新功能部件的安全性，应预先保存当前配置的备份。这将使得在对新功能部件或已更新的功能部件不满意时，可以很容易返回到原先的配置。保存当前配置的步骤如下。

① 在 Eclipse 工作台的上菜单中选择“帮助”→“软件更新”→“更新管理器”，这将打开“安装 / 更新”透视图。

② 在“安装配置”视图中，展开“配置历史记录”项并选择最新的条目，即最上面的那个条目，然后在上下文菜单中选择“保存”（或在对应的预览视图中单击“保存”按钮）。

③ 在“安装配置”视图中展开“已保存的配置”项以查看新保存的配置的条目。展开“已保存的配置”时将显示功能部件的位置。

④ 要将此已保存的配置的名称更改为可识别的名称，选择“已保存的配置”，并在上下文菜单中选择“属性”。

⑤ 在打开的“属性”对话框中，将“保留的配置的配置名”属性更改为更适当的名称。

⑥ 单击“确定”以使名称更改生效。

在“安装配置”视图中，“配置历史记录”中的条目的长度有一个上限，然而“已保存的配置”条目数没有限制，仅当在已保存的配置的上下文菜单中选择“除去”时才会删除该配置。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

7.4.3 恢复已保存的配置

可以恢复先前保存的配置或记录在历史记录中的任何配置，从而取消不成功的升级，实现还原。恢复已保存的配置不会删除本地计算机的功能部件和插件版本的文件；只是忽略它们并当做它们不存在。将已保存的配置恢复为当前配置的步骤如下。

① 在 Eclipse 工作台的主菜单中选择“帮助”→“软件更新”→“更新管理器”，这将打开“安装 / 更新”透视图。

② 在“安装配置”视图中，展开“已保存的配置”或“配置历史记录”项，并选择期望的配置，然后单击“恢复”按钮。

③ 当询问是否退出时单击“是”，并重新启动工作台以使更改生效。

④ 展开“当前配置”（和子代）以确认当前配置现在是否与先前保存的配置相匹配。

7.4.4 选取正在挂起的更改

每当启动 Eclipse 平台时，Eclipse 平台将会检测并记录已经对文件的已安装基本内容所作的任何更改，例如，功能部件和插件版本的出现或者消失。不会为现有工作区自动配置新的或已更新的功能部件；然而，在启动时将告诉用户存在更新并提供配置它们的机会。如果接受它们，则新的和已更新的功能部件版本就成为用于该工作区的功能部件和插件配置的一部分。如果不接受它们，则新的和已更新的功能部件版本对于该工作区就保持睡眠状态（禁用），但是稍后可选取以使用。选取正在挂起的更改的步骤如下。

① 当检测到新的更改时，将在启动时打开“配置更改”页面。或者，在 Eclipse 工作台的主菜单中选择“帮助”→“软件更新”→“正在审批的更改”，将打开“配置更改”页面，显示检测到的更改的列表。

② 选择并展开检测到的更改项，以查看相关联的功能部件版本。可以接受整个更改项，或选择个别功能部件。在某些新功能部件有错误的情况下，可能仅选择个别功能部件。

③ 选择要包括在配置中的项，忽略未选择的项，但是将保持挂起。

④ 单击“完成”以更新配置，或者单击“取消”以保持当前配置，挂起的更改继续保持挂起。如果已部分选择了挂起项，而不是所有功能部件，则单击“完成”按钮将标记已处理的项，并将从挂起更改列表中除去它们。

⑤ 当询问是否退出时单击“是”，并重新启动工作台以使更改生效。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 8 章　Eclipse 开发环境中的应用服务器 Tomcat

为了开发服务器端的应用程序，需要将 Eclipse 和应用服务器集成起来。Eclipse 可以和多种应用服务器共同工作，如 WebSphere、Weblogic、Oracle Application Server、Tomcat、JBoss 等。下面以目前流行的开放源代码的应用服务器 Tomcat 为例，讨论应用服务器的相关知识，以及 Eclipse 和应用服务器间的协作。

8.1　Tomcat 简介

Tomcat 是 Jakarta 的旗舰产品。Jakarta 的目标是提供一个基于 Java 平台的商业标准的服务器。目前，Tomcat 在严格意义上讲并不是一个完全的应用服务器，它只是一个可以支持运行 Servlet/JSP 的 Web 容器，Tomcat 4.1 内部实现了 Servlet 2.3 和 JSP 1.2 引擎。然而 Tomcat 也扩展了一些应用服务器的功能，如 JNDI、数据库连接池、用户事务处理等。Tomcat 被广泛地应用于中小规模的 Java Web 应用。

Tomcat 本身可以作为 Web 服务器，然而处理静态页面时，Tomcat 不如 Apache 效率高，Apache 作为最流行的 Web 服务器功能强大、高效，但并不支持 JSP 及 Servlet，所以通常的做法是将它们组合起来，让 Apache 处理静态页面，而把动态页面的请求交给 Tomcat 处理，发挥各自的优势。通过在 Apache 中加载整合模块和进行设置，Apache 就能够根据 URL，把不属于自己的请求转给 Tomcat。

Tomcat 是一个带有 JSP 环境的 Servlet 容器，Servlet 容器可以根据用户的行为可以管理和激活 Servlet 的运行。Tomcat 作为 Servlet 容器，有以下几种工作模式。

- 独立的 Servlet 容器

在这种模式下，Tomcat 内置有 Web 服务器的一部分，可作为独立的 Java Web 服务器。例如 Servlet 容器是 Java Web 服务器的一个部分。独立的 Servlet 容器是 Tomcat 的默认模式。

- 进程内的 Servlet 容器

在这种模式下，Servlet 容器作为 Web 服务器的插件和 Java 容器的实现。Web 服务器插件在内部地址空间打开一个 Java 虚拟机，使 Java 容器得以在内部运行。如有某个需要调用 Servlet 的请求，插件将取得对此请求的控制，并使用 JNI 将它传递给 Java 容器。进程内 Servlet 容器对于多线程的服务器非常合适，并且提供较高的运行速度，但伸

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

缩性不足。

- 进程外的 Servlet 容器

在这种模式下，Servlet 容器作为 Web 服务器的插件和 Java 容器的实现的结合，且运行于 Web 服务器之外的地址空间。Web 服务器插件和 Java 容器的 Java 虚拟机使用 IPC 机制，通常是 TCP/IP，进行通信。当一个调用 Servlet 的请求到达时，插件将取得对此请求的控制并将其传递给 Java 容器。进程外容器对请求的响应速度不如进程内容器，但进程外容器引擎具有更强的伸缩性和更高的稳定性。

Tomcat 既可作为独立的 Servlet 容器（主要是用于开发与调试），又可和其他 Web 服务器如 Apache、IIS 和 Netscape 等集成。

Tomcat 服务器由一系列的可配置的组件构成，其中核心的是 Catalina Servlet 容器，下面的 XML 代码描述了 Tomcat 中的组件关系：

```
<Server> <!--顶层元素，Tomcat 实例的顶层元素可包含一个或多个 Service-->
<Service> <!--连接器元素，包含一个 Engine 元素，以及一个或多个 Connector 元素，
              这些 Connector 共享同一个 Engine-->
   <Connector/> <!--实际和客户交互的组件-->
      <Engine><!--容器类元素，可以包含多个 Host 元素-->
         <Host><!--定义一个虚拟主机，它可以包含一个或多个 Web 应用-->
            <Context><!--使用最频繁的元素，每个 Context 代表运行在虚拟主机上的应用-->
            </Context>
         </Host>
      </Engine>
   </Service>
<Server>
```

8.2　Tomcat 的安装

Tomcat 具有开放源代码、跨平台等诸多特性，并且由于是自由软件，所以版本更新得很快。

8.2.1　Tomcat 的获取

从 Apache 的 Jakarta 项目的网站下载 Tomcat 服务器，目前最新的版本是 5.0.25，并有 5.0.27 的 β 版可供下载，下载的 URL 为：http://jakarta.apache.org/site/binindex.cgi。如 Tomcat 将运行在 Windows 操作系统中，可下载 zip 文件，5.0.25 版本的 Tomcat 的 zip 文件名为 jarkarta-tomcat-5.0.25.zip。

对于 Windows 操作系统，Tomcat 还有一种带安装器的安装文件，其文件后缀是.exe，如 jarkarta-tomcat-5.0.25.exe。

8.2.2　Tomcat 的安装和配置

Tomcat 需要在 JDK 上才能运行，因此安装 Tomcat 前首先需要安装 JDK。Tomcat

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

的安装操作只需将从 Apache 的 Jakarta 项目的网站上下载的 Tomcat 压缩文件解压到任一目录中。此外，需要设置两个环境变量：

JAVA_HOME=，其值为 JDK 的安装目录

CATALINA_HOME=，其值为 Tomcat 的安装目录

将 Tomcat 解压后，在 Tomcat 安装目录下可看到下列目录结构：

- /bin

bin 目录中包含在 Windows 平台和 Linux 平台上的启动和关闭 Tomcat 的脚本文件。

- /common/lib

存放 Tomcat 服务器和所有的 Web 应用程序都可以访问的 JAR 文件。

- /conf

包含不同的配置文件，包括 server.xml（Tomcat 的主要配置文件），以及为不同的 Tomcat 配置的 Web 应用程序设置的缺省值的文件 web.xml。

- /lib

包含 Tomcat 使用的 JAR 文件，.unix 平台此目录下的任何文件都被加到 Tomcat 的 classpath 中。

- /logs

Tomcat 存放日志文件的地方。

- /server/lib

存放 Tomcat 服务器所需的各种 JAR 文件。

- /server/webapps

存放 Tomcat 自带的系统管理和应用程序管理的 Web 工具。

- /shared/lib

存放所有 Web 应用程序都可以访问的 JAR 文件。

- /webapps

当发布 Web 应用程序时，默认情况下将 Web 应用程序放在此目录下。

- /works

Tomcat 把由 JSP 生成的 Servlet 放在此目录下。

运行安装目录下的“bin”目录中的脚本可以启动与关闭 Tomcat。在 Linux 下启动和关闭 Tomcat 的脚本分别是：

```
bin/startup.sh
bin/shutdown.sh
```

在 Windows 下启动和关闭 Tomcat 的脚本分别是：

```
bin\startup
bin\shutdown
```

在安装的时候，如果选择带安装器的安装文件，可将 Tomcat 安装为服务，安装程序可以把 Tomcat 自动添加到 Windows XP 或者 Windows 2000/2003 的服务中，服务名称是：Apache Tomcat 后接版本号，如 Apache Tomcat 5.0.25。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

假如将 Tomcat 安装为服务，用户可以在控制面板的服务中找到它，通过服务的图形界面启动 / 停止。用户也可以使用命令：net start “服务名”（如 net start “Apache Tomcat 5.0.25”）来启动 Tomcat 服务器；类似地，用户可使用命令：net stop“服务名”（如 net start “Apache Tomcat 5.0.25”）来停止 Tomcat 服务器。

如果在自动安装的时候没有选择安装服务，可以以后通过在命令行执行命令：tomcat -install “服务名” 来安装。

Tomcat 的配置基于两个配置文件：

- server.xml

该文件是 Tomcat 的全局配置文件，每个 Tomcat 组件在 server.xml 文件中对应一种配置元素。server.xml 完成两个目标。

① 提供 Tomcat 组件的初始配置。

② 说明 Tomcat 的结构和含义，使得 Tomcat 通过实例化组件完成起动及构建自身。

- web.xml

web.xml 在 Tomcat 中配置不同的关系环境。web.xml 是一个基于 XML 的部署描述符文件，Web 应用程序通过该文件来描述其部署信息。

Tomcat 提供了两个 Web 界面的管理工具，URL 分别是：

http://localhost:8080/admin/index.jsp

http://localhost:8080/manager/html

前一个管理工具配置系统管理，后一个管理工具配置应用程序的管理。在启用这两个管理工具之前，首先需要配置管理员用户和相应的口令。在 Eclipse 中可设置应用管理用户，本章将在后面的小节介绍。此处首先介绍系统管理的配置。

Tomcat 在 4.1 版本时首先引入了系统管理的 Web 管理工具，功能包括管理上下文、数据源、用户、用户组，像初始化参数等。

使用 Web 管理工具首先需要在 Tomcat 中配置具有系统管理员权限的用户，操作步骤如下。

① 导航到 Tomcat 安装目录下的 conf 目录。

② 选中该目录下的 tomcat-users.xml 文件，并使用文本编辑器打开该文件。

③ 在文件中添加下面的一行内容（假设本例中用户名为：admin，相应的口令为：admin123）：

```
<user name="admin" password="admin123t" roles="admin"/>
```

④ 保存并关闭 tomcat-users.xml 文件。

⑤ 重新启动 Tomcat。

经过以上步骤，Tomcat 中已添加了一个名为 admin、口令为 admin123 的管理员用户。类似地，可以设置任意名称的用户和对应的口令。

有关 Tomcat 访问数据库的配置的相关内容，本书将在“Eclipse 环境下的 MySQL 数据库应用开发”一章中进行介绍。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

8.2.3 Tomcat 的安装测试

安装并启动 Tomcat 后，可以在外部浏览器中测试 Tomcat 是否正常。在浏览器的 URL 中输入：http://localhost:8080，如 Tomcat 运行正常，则浏览器的显示如图 8.1 所示。

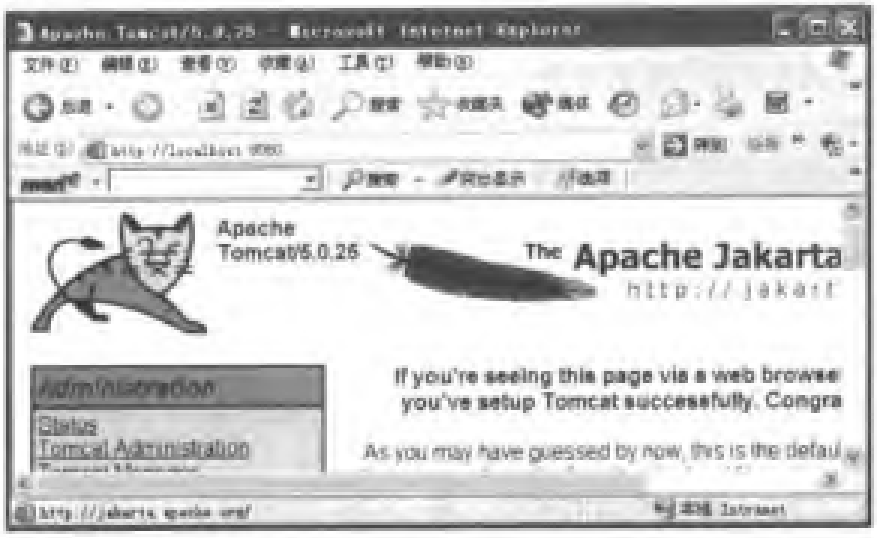


图 8.1 在浏览器中测试 Tomcat

如已设置了系统管理用户，访问 http://localhost:8080/admin 将可看到一个登录界面，如图 8.2 所示。



图 8.2 管理工具的登录界面

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Tomcat 的系统管理工具应用了基于容器管理的安全机制，并采用了 Jakarta Struts 框架。一旦作为“admin”角色的用户登录管理界面，将能够使用这个管理界面配置 Tomcat。登录后的系统管理用户界面如图 8.3 所示。

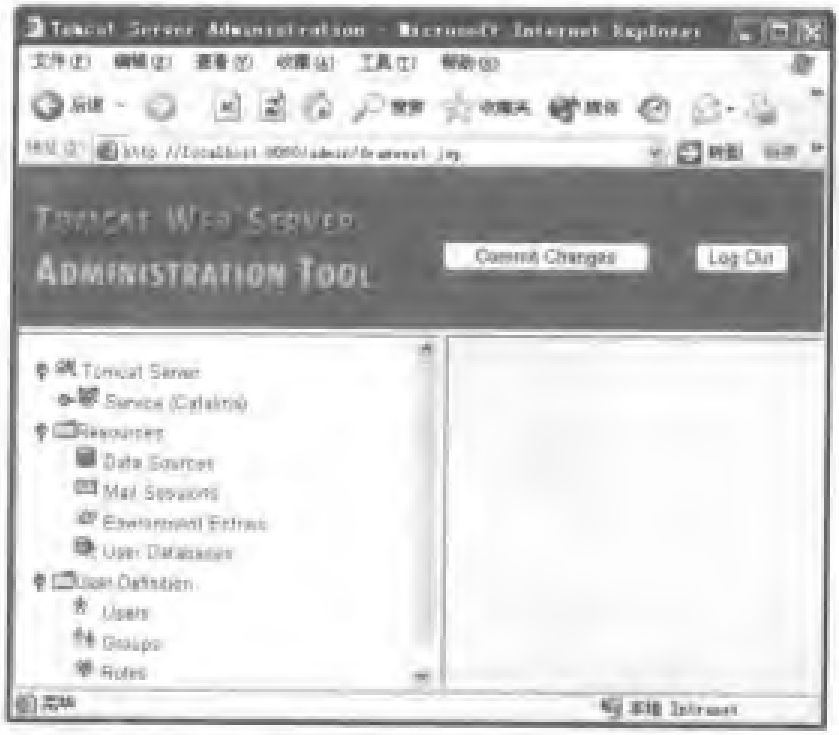


图 8.3 Tomcat 系统管理工具的界面

8.3 Sysdeo Tomcat 插件的安装和配置

Eclipse 是一个非常简单易用的集成开发环境环境，它具有很多特性，可以帮助程序员快速编写并调试 Java 程序。加上 Tomcat 插件之后，这个集成开发环境就是管理整个 Web 项目（包括 HTML 和 JSP 页面、图标和 Servlet 等）的一个非常优秀的工具。

从 Sysdeo 的 Web 站点下载 TomcatforEclipse 插件。Eclipse Tomcat 插件可以帮助程序员更好地组织并集成 Java 与 Web 开发项目。该插件目前最新的发布版本为 V221，也可下载 V3 的 a 版，下载的 URL 为：http://www.sysdeo.com/eclipse/tomcatPlugin.html。V221 版本的 Sysdeo Tomcat 插件的文件名为 tomcatPluginV221.zip。将该压缩文件解压缩到 Eclipse 安装目录下的\eclipse\plugins 子目录中，即可完成该插件的安装。

Sysdeo Tomcat 插件的安装后，即可对其进行配置，具体的配置步骤如下。

① 启动 Eclipse。

② 在 Eclipse 中，选择“窗口”→“首选项”（“Windows”→“Preferences”），打开设置窗口。

③ 在窗口左侧的“对象树”中选择“Tomcat”（在安装 Sysdeo Tomcat 插件之前，对象树中没有此对象），则在首选项窗口的右侧面板中，可以设置 Tomcat，如图 8.4 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理



图 8.4 Tomcat 的设置

④ 在 Tomcat 的设置窗格中，首先选择 Tomcat 的版本号。

⑤ 在 Tomcat 的设置窗格中，将“Tomcat home”设置为 Tomcat 文件的路径（例如对于 5.0.25 版本的 Tomcat，“Tomcat home”路径为：\Tomcat 的安装路径\jakarta-tomcat-5.0.25）。

单击“Tomcat Home”对话框边上的“浏览”按钮，浏览刚才解压的 Tomcat 的根目录，然后单击“确定”按钮。配置文件会自动被选中，并在对话框中添上相应的内容。

⑥ 在 Tomcat 的设置窗格中，将“Perspective to switch when Tomcat is started”设置为“Java”（默认的初始值是“None”）。

⑦ 在“首选项”窗口中选择“Tomcat”选项的子选项“JVM Settings”，如图 8.5 所示。

⑧ 在“JVM Settings”窗格中设置“Classpath”，单击“Zar/Zip”按钮，导航到 JDK 的安装目录下的 lib 子目录，选择 lib 子目录中的 tool.jar，然后单击“打开”按钮，将该文件添加到 Class 窗格中。

Tomcat Eclipse 插件要正常工作所必需的一个 SDK，这个 SDK 未必是 SunSDK，但是必须是一个 SDK，而不能是一个 JRE，因为这样不能正常工作（Eclipse 自身可以基于 JRE 运行）。为了让 Tomcat 能够与 Eclipse 一起正常工作，在 SDK 中必须有一个 Java 编译器。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 8.5 设置 Java 虚拟机

在应用程序开发中，当编译 JSP 文件时，将用到 tool.jar。

⑨ 在首选项窗口左侧的“对象树”中选择“Java”下的子选项“类路径变量”，确认 Tomcat 的路径已添加在其中，如没有则添加该路径。

⑩ 单击“确定”按钮，完成 Tomcat 在 Eclipse 中的设置。

⑪ 选择“窗口”菜单，然后选择“定制透视图子菜单”，“定制透视图”窗口如图 8.6 所示。

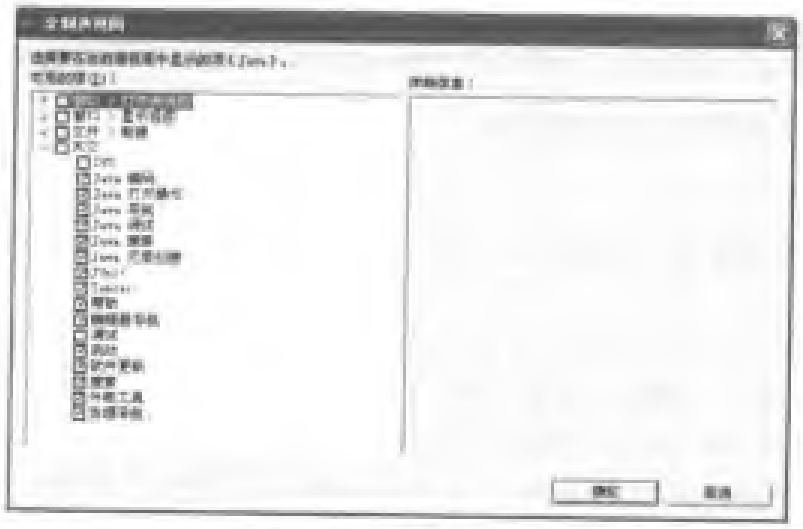


图 8.6 定制透视图窗口

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在“定制透视图”窗口的对象树中，选择“其他”项，并选中“Tomcat”子选项。

⑫ 单击“确定”按钮，则在 Eclipse 的 Java 透视图中将出现“Tomcat”菜单和“Tomcat”工具栏按钮，如图 8.7 和图 8.8 所示。



图 8.7 Tomcat 菜单



图 8.8 Tomcat 工具栏按钮

⑬ 使用“Tomcat”菜单或“Tomcat”工具栏按钮，将可启动或停止 Tomcat 的运行。相关的运行信息将显示在控制台中，如图 8.9 所示。

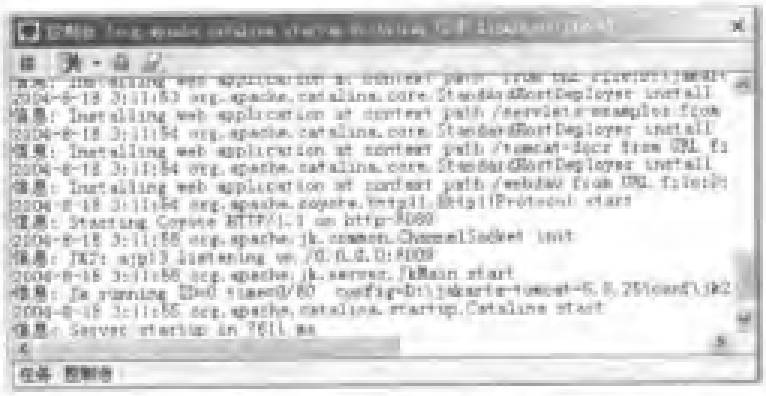


图 8.9 在 Eclipse 的控制台视图中显示的 Tomcat 的启动信息

Tomcat 的 Web 应用程序管理工具是一个比系统管理工具更为简单的管理工具，能执行一些简单的 Web 应用任务。

与 Web 系统管理工具一样，使用 Web 应用程序管理工具首先需要在 Tomcat 中配置具有应用程序管理员权限的用户。添加应用程序管理员用户操作步骤与添加系统管理员的步骤类似，惟一的差别是在 tomcat-users.xml 文件中添加：

```
<user name="manager" password="deep_dark_secret" roles="manager"/>
```

而不是配置系统管理员的添加内容：

```
<user name="admin" password="admin123t" roles="admin"/>
```

另一个更直观的配置方式是在 Eclipse 中直接配置 Tomcat 应用程序管理员用户和口令，具体步骤如下。

① 在 Eclipse 工作台主菜单中选择“窗口”→“首选项”，打开“首选项”窗口。

② 在首选项窗口左侧的对象树中展开“Tomcat”项，并选中其中“Tomcat Manager App”，右侧则相应出现 Tomcat 管理员用户和口令窗格，如图 8.10 所示。

③ 在“ManagerApp username”文本框中输入管理员用户名，在“ManagerApp password”文本框中输入对应的口令。

④ 单击“Add user to tomcat-users.xml”按钮，然后单击“确定”按钮。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 8.10　配置 Tomcat 管理员用户和口令

这样就完成了 Tomcat 应用程序管理员的配置。重新启动 Tomcat 以后，就可以使用所配置的用户来登录应用程序 Web 管理工具 http://localhost:8080/manager/html，如图 8.11 所示。从而可通过 Web 方式进行 Tomcat 的 Web 应用程序进行配置和管理。

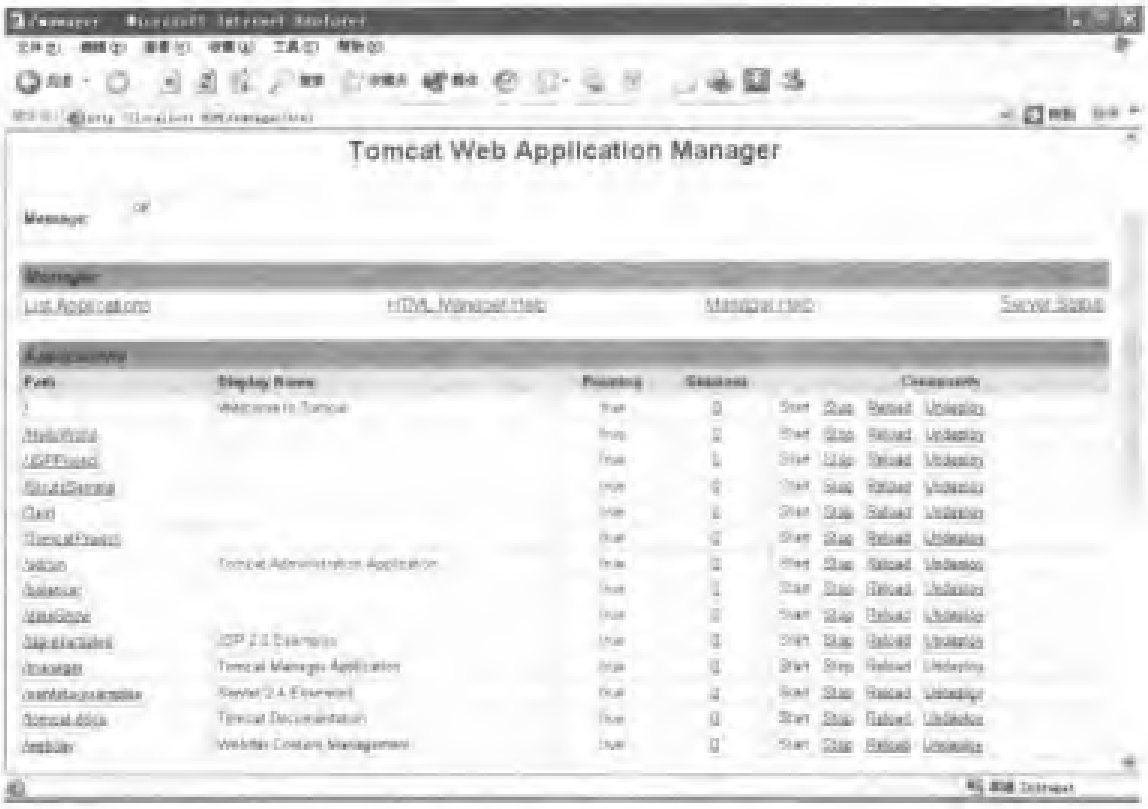


图 8.11　Tomcat 的应用程序管理工具

开发者可以在没有系统管理特权的基础上，使用应用程序管理工具部署新的 Web 应用程序，以用于测试。假设我们有一个新的 Web 应用程序位于/home/user/hello 下，并且想把它安装到/hello 下，为了测试这个应用程序，可以在“Context Path”中输入“/hello”

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

作为访问时的路径；在“XML Configuration file URL”中输入 XML 配置文件；在“WAR or Directory URL”中输入“file:/home/user/hello”，如图 8.12 所示。

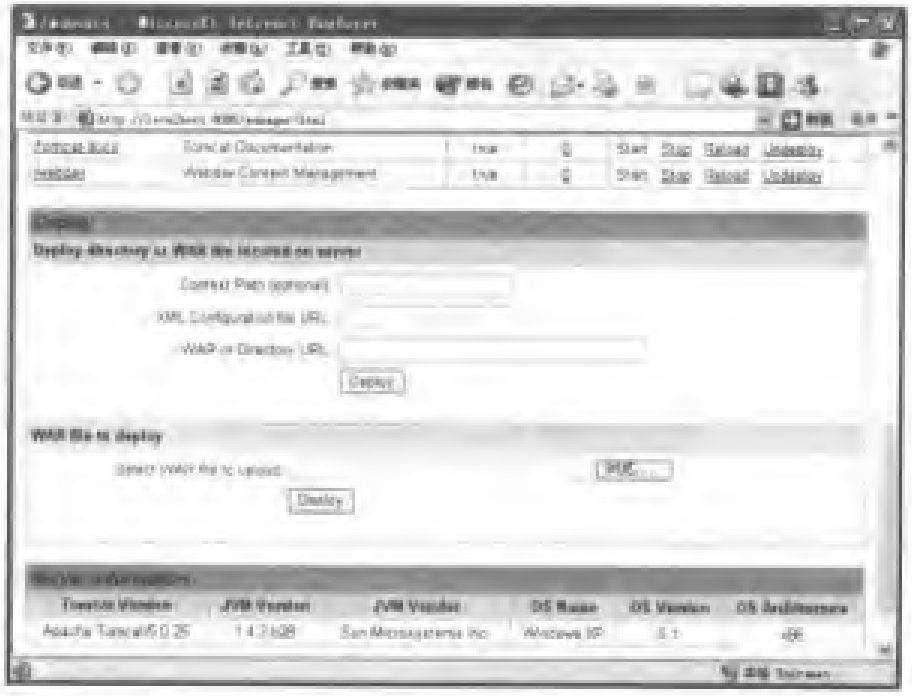


图 8.12　部署 Web 应用程序

开发者还可以使用 Web 应用程序管理工具停止、重新启动、移除，以及重新部署一个 Web 应用程序。当有用户尝试访问被停止的应用程序时，将看到一个 503 的错误：

```
"503 - This application is not currently available"
```

移除一个 Web 应用程序，只是指从 Tomcat 的运行拷贝中删除了该应用程序，如果重新启动 Tomcat，被删除的应用程序将再次出现。因此，如要完全删除该应用程序，需要在文件系统中删除。

8.4　服务器配置文件 server.xml

Tomcat 服务器的安装目录下的/conf/server.xml 文件是整个 Tomcat 服务器的配置文件，Tomcat 的所有组件都在其中配置。每个 Tomcat 组件在 server.xml 文件中对应一种配置元素。下面是以 XML 形式展示的各种 Tomcat 组件之间的关系。

```
<Server>
    <Service>
        <Connector/>
        <Engine>
            <Host>
            <Context>
            </Context>
            </Host>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
        </Engine>
      </Service>
</Server>
```

顶级元素<Server>是整个配置文件的根元素，而<Service>代表与一个引擎（Engine）相关联的一组连接器（Connectors）。Service 是一组 Connector 的集合，它们共用一个 Engine 来处理所有 Connector 收到的请求。连接器代表外部客户端程序之间的接口。外部客户端程序向特定的 Service 发送请求，并接收响应。Engine 用来处理 Connector 收到的 HTTP 请求，它将请求和自己的虚拟机进行匹配，并把请求转交给对应的 Host 来处理。容器代表一些组件，这些组件的功能是处理进来的请求，生成对应的响应。引擎处理一个 Service 的所有请求，主机（Host）处理一个特定虚拟主机的所有请求。Context 处理某个特定 Web 应用程序的所有请求。

下面将以一个 server.xml 文件为例，介绍文件的各个组成部分，并在文件中以注释的形式解释各个条目的含义。

```
<!-- port 指定一个端口，这个端口负责监听关闭 tomcat 的请求。在本例中，该端口为 8005。
     shutdown 指定向监听关闭的端口发送的命令字符串，在本例中，如果接收到"SHUTDOWN"字
     符串则关闭服务器。
 -->
<Server port="8005" shutdown="SHUTDOWN" debug="0">

 <!-Listener 指定监听进程
 -->
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"
  debug="0"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycle
  Listener" debug="0"/>

<!-全局 JNDI 资源
-->
  <GlobalNamingResources>
    ...
  </GlobalNamingResources>

      <!-name 指定 Service 的名字。在本例中为"Tomcat-Standalone"
      -->
        <Service name="Tomcat-Standalone">

     <!-className 指定实现 org.apache.catalina.Connector 接口的类。在本例中，类名
       为 org.apache.coyote.tomcat4.CoyoteConnector。
       port 指定服务器要创建的端口号，并在这个端口监听来自客户端的请求。在本例中，在端
       口号 8080 处侦听来自客户端浏览器的 HTTP1.1 请求。
       minProcessors 指定服务器启动时创建的处理请求的线程数。在本例中该 Connector 先
       创建 5 个线程等待客户请求，每个请求由一个线程负责。
       maxProcessors 指定最大可以创建的处理请求的线程数。在本例中，当现有的线程不够服
       务客户请求时，若线程总数不足 75 个，则创建新线程来处理请求。
       acceptCount 指定当所有可以使用的处理请求的线程数都被使用时，可以防到处理队列中
       的请求数，超过这个数量的请求将不予处理。在本例中，当队列中请求数超过 100 时，后来
       的请求返回 Connection refused 错误。
       redirectport 指定服务器正在处理 HTTP 请求时收到一个 SSL 传输请求后重定向的端口
       号。在本例中，当客户请求是 https 时，把该请求转发到端口 8443。
       connectionTimeout 指定超时的时间数，以毫秒为单位。在本例中为 20000 毫秒。
     -->
        <Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
                port="8080"
                minProcessors="5" maxProcessors="75" acceptCount="100"
                enableLookups="true"
                redirectPort="8443"
                debug="0"
                connectionTimeout="20000"
                useURIValidationHack="false"
                disableUploadTimeout="true" />

        <!-- name 指定 Engine 的名字。在本例中，Engine 名为"Standalone"。
             defaultHost 指定处理客户端请求的默认主机名，在<Engine>的<Host>子元素中
             必须定义这一主机。在本例中，默认虚拟主机是"localhost"。
        -->
          <Engine name="Standalone" defaultHost="localhost" debug="0">

        <!-- className 指定 logger 使用的类名，此类必须实现 org.apache.catalina.
             Logger 接口。在本例中，类名为"org.apache.catalina.logger.FileLogger"
        -->
          <Logger className="org.apache.catalina.logger.FileLogger" .../>

      <!--className 指定 Realm 使用的类名，此类必须实现 org.apache.catalina.Realm
          接口。在本例中为"org.apache.catalina.realm.UserDatabaseRealm"。
        -->
         <Realm className="org.apache.catalina.realm.UserDatabaseRealm" .../>

          <!-- name 指定主机名。在本例中，主机名为"localhost"。
               appBase 指定虚拟主机的目录。可以指定绝对目录，也可以指定相对于 Tomcat 安
               装目录的相对目录。如果此项没有设置，默认值是 Tomcat 安装目录下的/webapps
               子目录。在本例中为 Tomcat 安装目录下的/webapps 子目录。
               unpackWARs 如果为 true，则 Tomcat 会自动将 WAR 文件解压，否则不解压，直
               接从 WAR 文件中运行应用程序。在本例中，该值为 true，即自动将 WAR 文件解压。
          -->
            <Host name="localhost" debug="0" appBase="webapps" unpackWARs=
             "true" autoDeploy="true">

                <Logger className="org.apache.catalina.logger.FileLogger" .../>

                 <!-- path 表示此 Web 应用程序的 URL 的前缀，这样请求的 URL 为 http://
                      localhost:8080/path/****。在本例中 Context 的路径名是""，故
                      该 Context 是该 Host 的默认 Context。
                      docBase 指定应用程序的路径或者是 WAR 文件存放的路径。在本例中，
                      该 Context 的根目录是 webapps/mycontext/
                  -->
                     <Context path="" docBase="mycontext" debug="0"/>

                    <Context path="/wsota" docBase="wsotaProject" debug="0"/>
                  </Host>
              </Engine>
        </Service>
</Server>
```

<Context>元素代表了在 Tomcat 服务器主机上的单个 Web 应用，即每一个 Web 应用对应着 server.xml 文件中的一个条目。Tomcat 严格遵守 Java Servlet 2.3 和 JavaServer Pages 1.2 规范。Tomcat 可以和目前主流的 HTTP 协同工作，实现稳定可靠的 Web 应用。本书将在第 9 章介绍结合 Tomcat 和 Eclipse 开发 Web 应用程序的相关技术。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 9 章　Eclipse 下的 Web 应用开发

Java Web 应用程序是由一组 Servlet、JSP 页面、HTML 页面、Java 类及其他可以被绑定的资源组成。所谓绑定在这里时指，这些资源在相同的 Web 上下文中，并且在许多情况下将直接或间接地相互调用。

9.1　Servlet

传统的 Web 服务器只能提供静态 HTML 网页的服务，而应用服务器通过在 Java 虚拟机上运行 Java 代码，极大地扩展了服务器的功能。因此可以将 Servlet 看做是那些只能提供静态内容的传统 Web 服务器功能的一种扩展。Servlet 可以在执行时产生动态内容，因此开发人员可以通过创建新的 Servlet 来添加新的功能。

9.1.1　Servlet 简介

Servlet 是用 Java 编写的，与平台无关的服务器端 Web 组件，因此 Servlet 是在服务器端运行的 Java 程序。这与 Applet 不同，Applet 是运行在浏览器端的。Servlet 是独立于平台的 Java 类。Servlet 规范是 J2EE 规范的一部分，因此 Servlet 应用程序具有很好的可移植性。

在 Servlet 技术诞生之前，通常使用 CGI 脚本提供服务器端的动态服务。Servlet 扩展了 Web 服务器和应用服务器的功能，并由 Web 容器进行管理。Web 容器可以控制对 Servlet 的访问，从而提高了 Servlet 的安全性，并可作为整个安全体系结构如 SSO（单击登录）的一部分。

CGI 脚本程序响应每一个新的 HTTP 请求都需要创建和执行一个新的进程，从而需要耗用大量的系统资源。请求数越多意味着创建进程所耗费的时间越多，并最终会大大地降低服务器的性能、减慢响应速度。然而，Servlet 针对 HTTP 请求提供了多线程机制，对于每一个请求，只需创建和执行一个新的线程，即只需加载 Servlet 一次，因此大大降低了系统开销，从而降低了对系统资源的耗用。

Servlet 类提供了针对 HTTP 请求的输入数据如参数或表单数据等的分析和预处理功能，并且支持会话跟踪和 Cookie。因此，Servlet 能够更好地支持 Web 端的动态服务。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

9.1.2 Servlet 的处理流程

Servlet 通常在应用服务器的 Java 虚拟机中运行。对于 Servlet，通常 Web 容器使用 HTTP 请求—响应协议响应来自 HTML 请求的用户事件，然后动态构建 HTML 响应返回给用户。

如图 9.1 所示，浏览器向 Web 服务器发出 HTTP 请求，Web 服务器插件将判断请求的是诸如标准的 HTML 网页文件、图像文件或者声音文件之类静态内容，还是动态内容。Web 服务器插件通常安装为 Web 服务器的一部分，但通常在安装应用服务器时进行 Web 服务器插件的安装。如客户端请求是对静态资源的访问，则 Web 服务器插件将请求转发给 Web 服务器；如是对动态资源的请求，如 Servlet，Web 服务器插件则将请求转发给 Web 容器。

应用服务器内的 Web 容器处理和 Servlet 间的通信，传递对象的请求和响应。因为 Servlet 是使用标准 Java 接口编写的 Java 代码，所以 Servlet 可以与其他的类或对象合作完成另外的工作，例如执行一个访问数据库的查询。此时 Servlet 得到客户请求的结果。因为最终将要通过浏览器显示返回结果，所以 Servlet 把返回结果格式化为浏览器可以理解的 HTML 语言。因此，Servlet 直接或者间接地动态产生送回到浏览器的 HTML 响应。

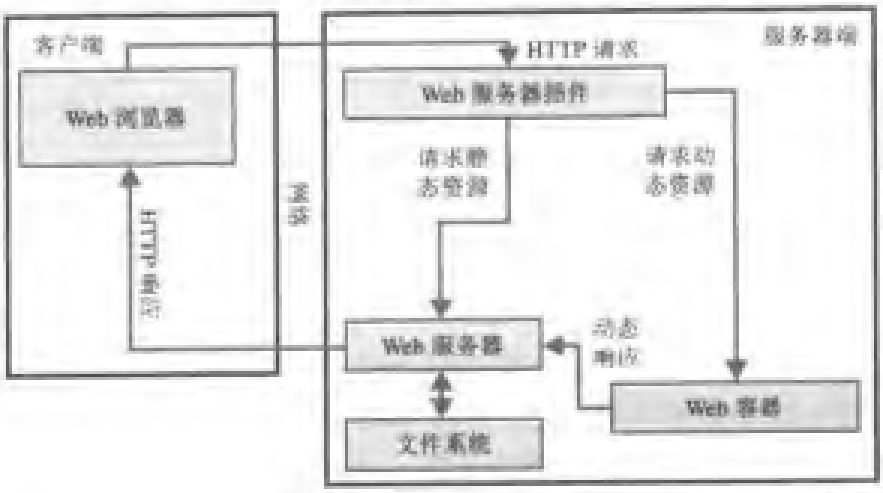


图 9.1 HTTP 请求的处理流程

为了能处理客户端和服务器之间的通信，Sevelet 实现了一个通用的请求响应范例。为处理客户和服务器之间的请求响应消息，Java Servlet API 定义了相关的标准接口。如图 9.2 所示，客户端请求到 Servlet 处理间的处理流程如下。

① 客户端向服务器发送请求，请求的 URL 中包含标识 Servlet 的信息。

② Web 服务器将请求转发给 Servlet 引擎，Servlet 类的实例驻留在 Servlet 引擎中。

③ Servlet 引擎调用 Servlet 的 Service 方法，Service 方法是所有 Servlet 中都要定义的方法。

④ Servlet 动态构建一个响应，并将它传递给 Web 服务器。

⑤ Web 服务器将响应返回给客户端。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 9.2 Servlet 的处理流程

9.1.3 Java Servlet API

Web 应用程序的 J2EE 基础是 Java Servlet API。Java Servlet API 是一组定义了 Web 客户端和 Web Servlet 之间标准接口的 Java 类。客户端的请求发送给 Web 服务器。然后，Web 服务器通过这个接口调用 Servlet 为这个请求服务。

Java Servlet API 是标准的 Java 扩展 API。该 API 由以下两个包组成：

- javax.servlet
- javax.servlet.http

包 javax.servlet 包含支持通用的独立于协议的 Servlet 的类，即 Servlet 能用于多种协议。Servlet 框架的核心是 javax.servlet.Servlet 接口，所有的 Servlet 都必须实现这一接口。一个特定的 Servlet 对象是实现 javax.servlet.Servlet 接口的 Java 类的实例。Servlet 接口类是 Java Servlet API 的集中抽象。这个类定义了 Servlet 必须实现的方法。在 Servlet 接口中有 3 个方法代表了 Servlet 的生命周期：

- init()

负责初始化 Servlet 对象。

- service()

负责响应客户端的请求。

- destroy()

当 Servlet 对象退出生命周期时，负责释放占用的资源。

包 java.servlet.http 扩展了基础包的功能，对 HTTP 协议进行了进一步的增强。

9.1.4 Servlet 生命周期

Servlet 具有明确的生命周期，包括实例化、初始化、处理服务调用、销毁等，Java 虚拟机按照预先定义的方式执行 Servlet。Web 容器负责 Servlet 生命周期服务。单个的 Servlet 的生命周期如图 9.3 所示。

当应用服务器接受请求时，首先确定所请求的 Servlet 是否已经加载到内存中，Servlet 是否可用。如果这个 Servlet 没有被加载，应用服务器将加载这个 Servlet 类，并调用 Servlet 默认的构造函数创建该 Servlet 的一个实例。接下来，应用服务器调用 Servlet 的 init()方

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

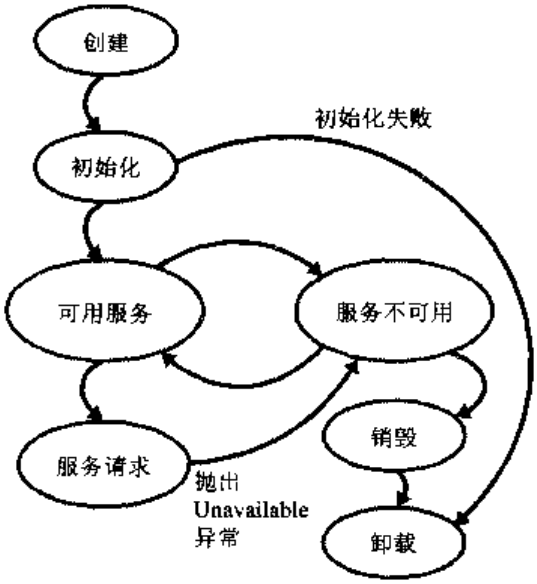


图 9.3　Servlet 的生命周期

法。开发人员用 init 方法来完成 Servlet 的初始化任务，例如加载 Servlet 需要的资源。当 init 方法完成后，Servlet 就可以处理请求了。具体地说，Web 容器按照以下顺序创建和初始化 Servlet 实例。

① 使用默认（没有参数）的构造函数（constructor）创建 Servlet。

② 创建 ServletConfig 实例。

③ 调用 Servlet 对象的 init()方法，ServletConfig 对象作为参数。

④ 成功初始化后，Servlet 的状态将标记为可用；如初始化过程失败，Servlet 将标记为不可用。

响应请求时，Web 容器调用 Servlet 的 service()方法。HttpServlet 还为 HTTP 请求方法，如 GET、POST、PUT 等，定义了一组特定的请求处理程序 doGet()和 doPost()。当 Web 客户端发出一个 HTTP GET 请求时，默认的 service()方法将调用 doGet()方法。当 Web 客户端发出一个 HTTP POST 请求时，默认的 service()方法将调用 doPost()方法。

当 service()方法完成时，请求的处理也就结束了。应用服务器在内存中保留 Servlet 实例来处理将来的请求。当收到另一个请求时，应用服务器不需要重新开始一个进程的生命周期。相反，因为 Servlet 仍然在内存中存在，应用服务器只需再次简单地调用 service()方法。

实际上，对于多个并发请求，应用服务器几乎是同时为每个请求调用 service()方法。当多个请求同时发生时，应用服务器使用多线程技术来处理请求，每个请求将拥有自己的线程。请求同一个 Servlet 将使用应用服务器内存中保存的同一个 Servlet 实例。

应用服务器并不在内存中永久保持 Servlet 的实例。应用服务器的管理员可以设定一个超时值。超时值表明应用服务器等待多久后注销 Servlet 的实例。超时值在每一次新请求后被重置。所以只要有新的请求到达，应用服务器就在内存中保持着 Servlet。一旦超时，应用服务器将调用 Servlet 的 destroy()方法，并在内存中移去它的实例。如果 Servlet

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

被注销后，又有新的请求到达，则整个生命周期过程从头开始。

9.1.5 一个简单的 Servlet 程序的示例

下面将通过一个简单的 Servlet 来讨论 Java Servlet API 以及 Servlet 的相关知识。

```
package simple.servlet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletException;
import java.io.IOException;
import java.io.PrintWriter;

public class SimpleServlet extends HttpServlet {

  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {

      String browser = request.getHeader("User-Agent");
      response.setStatus(HttpServletResponse.SC_OK); // default
      response.setContentType("text/html"); // default
      PrintWriter out = response.getWriter();

      out.println("<HTML><HEAD><TITLE>Simple servlet");
      out.println("</TITLE></HEAD><BODY>");
      out.println("Browser details: " + browser);
      out.println("</BODY></HTML>");

  }
}
```

一个特定的 Servlet 对象是实现 javax.servlet.Servlet 接口的 Java 类的实例。一般情况下，该 Java 类将扩展抽象类 javax.servlet.http.HttpServlet，因此在 Servlet 程序中需要 import javax.servlet.http.HttpServlet。

HTTP 请求中包含的信息，无论是由用户专门输入的还是由浏览器自动提供的，都会作为实现 javax.servlet.http.HttpServletRequest 接口的对象传给 Servlet。在 Servlet 中，HTTP 输入流是封装在实现 javax.servlet.http.HttpServletRequest 接口的对象中的，该接口扩展了 javax.servlet.http.HttpServletRequest 接口。因此在 Servlet 程序中需要 import javax.servlet.http.HttpServletRequest。

响应请求时，Web 容器调用 Servlet 的 service()方法。抽象类 HttpServlet 实现了 service()方法，如下所示：

```
public void service(ServletRequest request, ServletResponse response)
    throws ServletException, IOException
```

在 Servlet 程序中，需要重载 doGet()（或 doPost()）方法用于处理 HTTP 的 GET 或 POST 请求。

GET 方法的请求处理程序如下所示：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
```

POST 方法的请求处理程序如下所示：

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
```

方法 service()、doPost()和 doGet()的签名都有两个参数。第一个参数是请求对象，Servlet 需要能够访问到来的 HTTP 流。请求对象 Servlet 提供所有的输入值。例如，HTML 表单数据、cookie 和请求 URL 等信息都可以在 Servlet 请求中获得。另外，Servlet 请求还包含环境信息，例如当前的会话设置和安全设置。第二个参数的类型是 HttpServletResponse，Servlet 响应是 javax.servlet.http.HttpServletResponse 接口一个实例的实现。因此，在 Servlet 程序中需要 import javax.servlet.http.HttpServletResponse。Servlet 响应为 Servlet 提供输出数据的方法。例如，Servlet 响应包含创建各种输入 / 输出（I/O）流发送数据到客户端的方法。Servlet 响应还包含存储信息的方法，或者在客户端（例如 cookie），或者使得浏览器显示另一个 URL（也称重定向）。请求对象和响应对象都能抛出 ServletException 类型的异常和 IOException 类型的异常。因此，在 servlet 程序中需要 import javax.servlet.ServletException 和 java.io.IOException。

在 Servlet 中需要做的首件事情就是处理 HttpServletRequest 参数中的头部信息、cookie、输入数据或表单数据等。在本例中，Servlet 读取 User-Agent 请求头部，它提供了有关用户浏览器的信息。然后在 Servlet 中设置 HttpServletResponse 对象的头部值，将状态设为 SC_OK。最后，Servlet 程序构建了 HTML 输出文档。

9.1.6 Eclipse 中的 Servlet 应用程序的开发

下面介绍结合 Eclipse 和 Tomcat 技术开发一个简单 Servlet 的例子。具体步骤如下。

① 在 Eclipse 工作台的主菜单中选择“文件”→“新建”→“项目”。

② 此时将打开“新建项目”对话框，如图 9.4 所示，在对话框的左侧选择“Java”，然后在右侧选择“Tomcat Project”，然后单击“下一步”按钮。

③ 此时将打开“New Tomcat Project”对话框，在对话框中输入项目名，这里可以假设为 HelloWorld，然后单击“下一步”按钮。

④ 在显示的对话框中选中“Can update server.xml file”选项。在默认情况下，该选项已选中，然后单击“完成”按钮。

⑤ 在项目的上下文菜单中选择“新建”→“类”，将出现“新建 Java 类”对话框。

⑥ 在对话框的名称字段中输入 HelloWorld，在超类中填入 javax.servlet.http.HttpServlet，然后单击“完成”按钮，一个 Servlet 就创建好了。

⑦ 在 HelloWorld.java 中重载 doGet 方法。HelloWorld.java 中的代码如下：

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
import java.io.PrintWriter;
import javax.servlet.ServletException;

public class HelloWorld extends HttpServlet {
public void doGet(HttpServletRequest request,HttpServletResponse response)
      throws IOException, ServletException {

response.setContentType("text/html");
PrintWriter writer = response.getWriter();
writer.println("Hello World!");
      }
}
```



图 9.4　创建 Web 应用项目

⑧ 将 Servlet 发布到 Tomcat。

9.2　JSP

JSP 是 Java Server Pages 的缩写，它是一种动态网页技术标准。JSP 技术是 Java 平台上用于编写包含诸如 HTML、DHTML、XHTML 和 XML 等含有动态生成内容的 Web 页面的应用程序开发技术。

9.2.1　JSP 简介

JSP 非常像 Servlet，能够生成动态页面。一般来说，生成的是 HTML 页面，但也可以生成 DHTML、XHTML 和 XML 页面。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

JSP 能够做到的任何事情 Servlet 也同样可以做到。那为什么不直接使用 Servlet 呢？主要原因是因为在处理显示问题方面，JSP 技术比 Servlet 技术具有更大的优点。

使用 Servlet 必须在 Servlet 中生成标记，而在当前的生成工具中，验证 Java 程序中的这些标记是否正确很困难。因为编译器仅仅将这些标记看做字符串，所以判断这些标记是否良定义或者是否正确是非常困难的，甚至是不可能的。随着标记越来越复杂，用 Servlet 生成这些复杂的标记越发困难。

在 Servlet 中使用硬编码的 HTML 语言作为请求响应的一个严重缺陷是，很难对 HTML 进行修改，每次修改都必须重新编译 Servlet。此外，另一个问题是 Web 应用程序的国际化问题。由于 HTML 是硬编码的，所以维护不同的语言很困难。判断用户的语言、所处的地区以及其他信息，然后再显示输出很困难。

Web 页面的内容包括静态和动态内容。JSP 技术可以方便地生成静态和动态的 Web 内容。静态内容仍然为标记，而脚本语言，通常为 Java，生成动态的内容。标记可很好地格式化内嵌的脚本语言生成的动态结果。使用 JSP 技术，Web 应用开发人员可以使用 HTML、XSL 等来设计与布局最终页面的显示，使用 JSP 标记或 Scriptlet 来生成页面上根据请求而变化的动态内容，从而实现了应用程序的内容与页面显示的分离。

为了对业务逻辑和表示逻辑进行很好的分离，使用 JSP 来专门处理表示逻辑的相关工作。JSP 技术封装了许多功能。JSP 可以访问所有的 J2EE 组件，包括 JavaBean、EJB、Servlet 等。如图 9.5 所示，标准的 JSP 标记能够访问和实例化 JavaBean 组件，设置或者检索组件属性等。

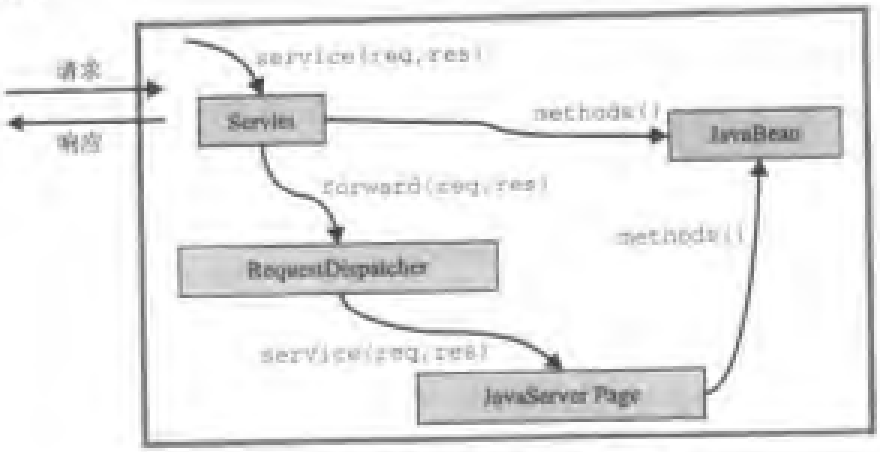


图 9.5　Web 应用程序中的 Servlet、JSP 和 JavaBean

9.2.2　JSP 的实现和运行方式

JSP 页面一般混合着 HTML 和 Java 代码，前者是静态的内容，后者则用来生成动态的内容。为了融合静态和动态的内容，需要生成、编译和执行 Java 代码。

JSP 规范是定义在 Java Servlet API 之上的标准扩展。因此，JSP 引擎一般都构建于 Servlet 引擎之上。如图 9.6 所示，当应用服务器收到对 JSP 页面的请求，或者当一个 Servlet 把对 JSP 页面的请求转发过来，应用服务器首先把 JSP 文件编译成 Java 源文件，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

然后将该源文件编译成一个类文件。执行该类文件将输出静态的和动态的内容。运行类文件将在浏览器上显示 JSP 页面。



图 9.6 JSP 的执行模型

如果改变了 JSP 文件的内容，应用服务器将重复这个转换和编辑的过程。否则，应用服务器将重复使用编译好了的类文件。

JSP 中 Java 接口的设计方式决定了 JSP 实际上是一个 Servlet（在运行期间表现得更像），但是 Servlet 却不是 JSP 页面。应用服务器将 JSP 页面编译为一个 Java 源文件，这个源文件中定义了一个 Java 类来实现 HttpJspPage 接口。

由 JSP 页面编译生成的 Java 源文件中包含一个_jspService 方法。这个方法与 Servlet 的 service 方法类似。对于 HTML 中的任何一个部分，_jspService 方法都调用 JspWriter 的一个方法，来输出结果的一部分。这个过程与 Servlet 调用 PrintWriter 非常类似。嵌入在 JSP 页面中的所有 Java 代码都以_jspService 方法中的交叉存取结束。所以，在输出静态内容期间，执行嵌入的 Java 代码并产生动态的内容。这样，静态内容和动态内容被混合到输出流中。

由于JSP页面是一个Servlet，应用服务器处理编译后的JSP类文件与其他任何Servlet没有什么不同，并与 Servlet 有着相同的生命周期。一旦载入，应用服务器把类文件保存在内存中以备响应未来的请求。换句话说，应用服务器通常创建生成的 Servlet 的一个实例，并由这个实例来响应多个请求。直到 JSP 页面的源文件被改变，应用服务器才重新编译该 JSP 页面。然后应用服务器卸载老的类文件，并将新的类文件加载到内存中。

9.2.3 JSP 的访问模型

JSP 可用于若干不同的应用体系结构中。相应地，对于 JSP 的访问有着不同的访问模型，主要有如下三种。

在第一种最简单的模型中，如图 9.7 所示，客户端浏览器直接请求 JSP 页面，JSP 页面自己生成被请求的内容，并构建 HTML 将结果返回给用户。

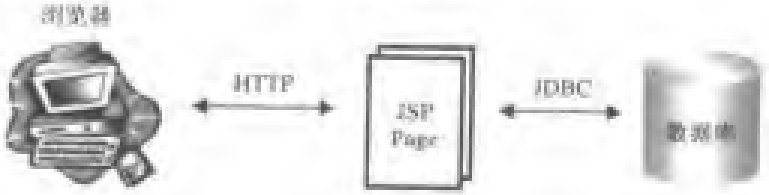


图 9.7 直接访问 JSP 页面的模型

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

使用这种模式可以快速开发出 Web 应用程序，但不适合应用在并发访问量很大的企业应用系统中。例如，因为每个客户端都必须建立或者共享一个到资源的连接，从而有可能建立大量的连接，例如大量的到数据库的连接，这将会影响整个系统的性能。

第二种模式中存在一个起着访问控制的 Servlet，如图 9.8 所示。在这种模式中，其访问过程如下。

① 客户端浏览器通过 Web 服务器直接对 Servlet 进行请求。

② Web 容器调用相应的 Servlet 的 service()方法。

③ Servlet 和相关的 JavaBean 交互，并由 Servlet 生成动态的内容。

④ Servlet 再将结果绑定到一个结果对象，并且调用 JSP 页面。

⑤ JSP 页面从该对象中访问动态内容，并将构建的内容如 HTML 文档返回给客户端浏览器。

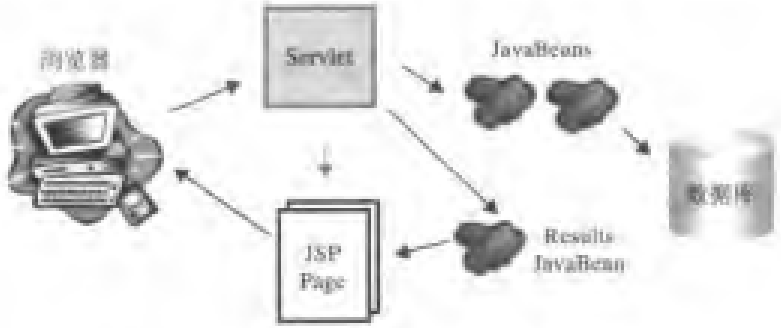


图 9.8　在具有控制器 Servlet 的结构中访问 JSP 的模型

这种方式创建了更多的可以在应用程序之间共享的可重用组件，并且可以作为整个应用系统的一部分来实现，但是在访问企业资源时，还存在可伸缩性问题。

第三种模式中，JSP 作为 J2EE 体系结构中的中间层次，如图 9.9 所示。

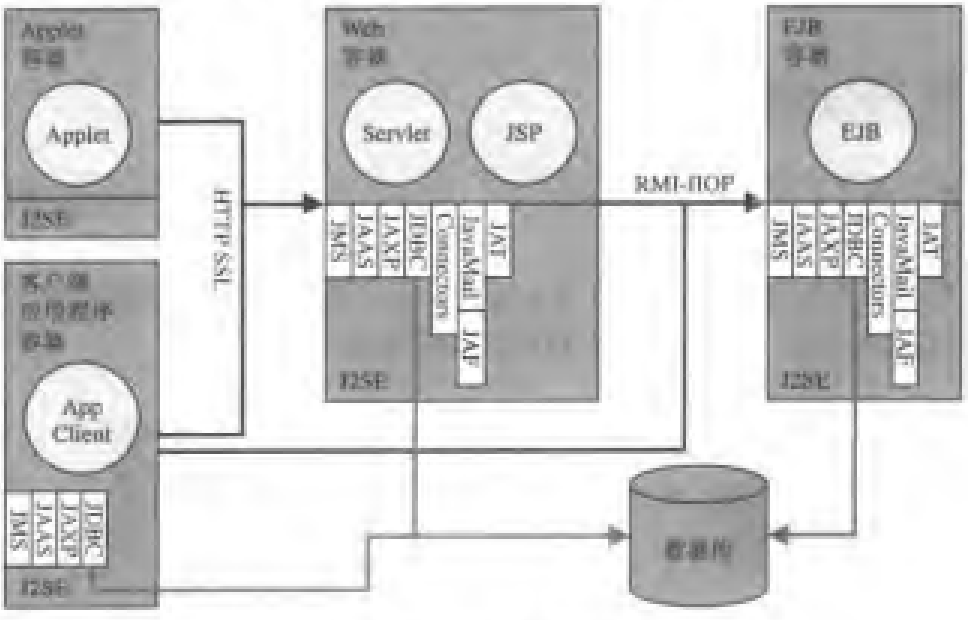


图 9.9　J2EE 体系结构

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在这种方式中，JSP 页面和后端资源通过 EJB 组件进行交互。在整个 J2EE 体系结构的应用系统中，各组件间的关系如图 9.10 所示。

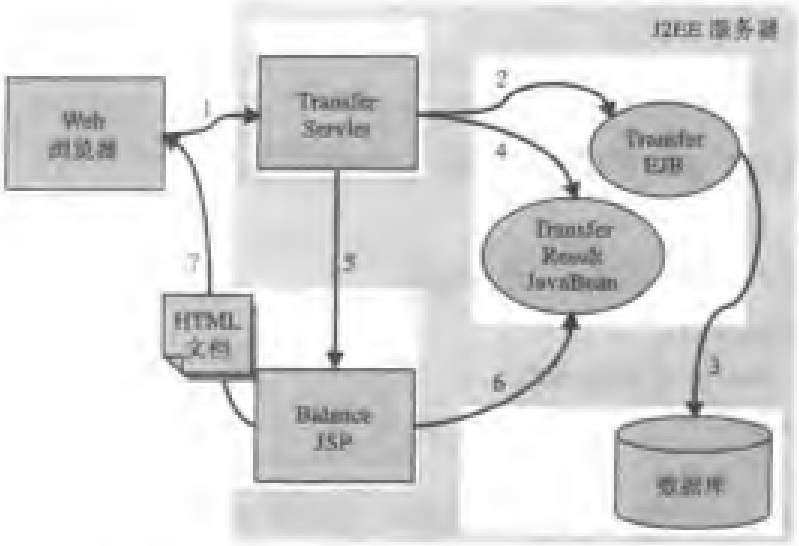


图 9.10　在 J2EE 体系结构中访问 JSP 的模型

9.2.4　JSP 文档结构和语法概述

JSP 文件与 HTML 和 XML 从外表上看没有很大的区别，只是多了一些使用 Java 语言编写的，JSP 引擎能够处理的标签。JSP 源文件是一种标准的带有特殊 JSP 标签的标记文件，如 HTML 和 XML 等。这个附加的 JSP 语法是为了在结果表示中提供动态内容。在很大程度上，JSP "标签" 直接对应了可执行的 Java 代码。下面是一个显示当前时间的简单 JSP 源代码的示例：

```
<HTML>
    <HEAD><TITLE>Simple JSP</TITLE></HEAD>
    <BODY><H1>Today's date and time are:
            <%= new java.util.Date() %>
        </H1>
    </BODY>
</HTML>
```

下面，我们将简要介绍 JSP 规范中的核心语法。

1. 注释

要编写一个良好的、易于交流的页面或者程序，添加适当的注释是必不可少的。在 JSP 页面中可以使用两类注释。

① 正规的 HTML 和 XML 注释。

该注释将出现在动态生成的，发送到客户端的文件中的注释，其语法形式如下：

```
<!-- 这是一个 HTML/XML 注释 -->
```

该类型的注释内容将不会在浏览器的页面中显示，然而通过查看源代码的方式可以

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

查看到该类注释。

② JSP 注释。

这种注释是对 JSP 页面本身的代码的注释，它不能被输出到客户端，其语法形式如下：

```
<%--这是一个 JSP 注释。--%>
```

这类注释仅存在于 JSP 源代码中，即 JSP 将忽略它。

2. 元素

JSP 规范定义了一组“标签”，分为三类：

- 伪指令（Directive）
- 脚本元素（Scripting element）
- 动作（Action）

（1）伪指令

伪指令是用于 JSP 引擎 / 编译器的指令，因此将会影响 Java 运行时类的生成。JSP 规范 1.2 定义了三种 JSP 伪指令：include、taglib 和 page。

变换时，伪指令 Include 在相应的位置静态地包括一个文件，例如：

```
<%@ include file="readme.html" %>
```

这个包括是在变换时完成的，而不是运行时。所以，如果包括的资源发生变化，并且包含它们的 JSP 需要了解那些变化，那么应该使运行时包括 action，而不是伪指令。再者，若资源不会发生变化，就该使用 Include 伪指令，因为它没有任何运行时影响。

第二个是 taglib 伪指令，编译页面时，变换器（Translator）将使用该伪指令所提供的标签库信息。使用 JSP 库提供的定制标签进一步划分了页面设计人员和 Java 编程人员这两种角色所要求的职责和技术。Java 编程人员创建定制标签库，而页面设计人员使用定制标签，页面设计人员无须掌握 Java 编程语言知识，就可以在 JSP 页包含动态内容。

这些库需要在其动作使用前先在 JSP 页中定义。例如，核心标签库在标签库指示中定义如下：

```
<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
```

第三个是 page 伪指令。这个伪指令有很多属性，这些属性向变换器说明所生成的类的不同特性。

（2）脚本元素

JSP 中定义了三种不同风格的脚本元素：

- 声明
- scriptlet
- 表达式

声明用来定义类一级的成员，如域和方法。它们最常用来许可 jspInit()和 jspDestroy()方法的执行。声明的语法为<%!…%>。XML 版本的声明标签是：

```
<jsp:declaration></jsp:declaration>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

scriptlet 标签用于在 JSP 页面中嵌入代码，一般是 Java 代码。scriptlet 和表达式都用来向 _jspService()方法体添加 Java 代码。在这个方法体中，有隐式和显式两种脚本对象（变量）。隐式对象只能使用。显式对象对应脚本变量并且可以在 scriptlets 和表达式中定义和赋值。

定义 Scriptlet 的语法是<%…%>，下面是 scriptlet 的一个语法示例：

```
<%
    Invoice inVoice = (Invoice)request.getAttribute("invoice");
    Out.println( invoice.getDescription() )
%>
```

scriptlet 标签的 XML 格式为：

```
<jsp:scriptlet></jsp:scriptlet>
```

JSP 表达式允许指定被显示的 Java 表达式。JSP 表达式很有用，通常可读性也比较好。表达式标签的 XML 格式是：

```
<jsp:expression></jsp:expression>
```

（3）动作

JSP 最后一个组标签是动作。Action 可分为两大类：在 JSP 规范中预定义的动作，也称为标准动作；通过标签库自定义的动作。在表 9.1 中，列举了标准动作：

表 9.1 JSP 标准 action

Action 标签	描 述
<jsp:include page="urlSpec" />	将在同一 Web 组件中的资源包含当前响应流
<jsp:forward page="urlSpec" />	把所有当前请求的运行时分派传送给同一 Web 组件中的资源
<jsp:useBean"id=" "scope="typeSpec> </jsp:useBean>	将 Java 对象的实例和一个声明的脚本变量（它的名字和 ID 的值相同）关联起来。该 Java 对象实例应在给定作用域内定义并可通过给定 ID 号使用
<jsp:setProperty>	允许 bean 的属性根据请求参数或是运行时表达式初始化。通常在 jsp:useBean 标签体中用到
<jsp:getProperty>	替换 bean 实例属性值，转换为字符串，成为供显示的隐式 out 对象
<jsp:plugin>	用来创建合适的客户端浏览器结构（OBJECT 或是 EMBED），会导致下载 Java Plugin（若需要的话），以及后面 applet 或是 JavaBean 组件的下载
<jsp:param>	和<jsp:plugin> action 一起使用，用来指定 applet 参数（名字，值对）
<jsp:params>	和<jsp:plugin> action 一起使用，块封装了<jsp:param>标签
<jsp:fallback>	和<jsp:plugin> action 一起使用，提供文本显示 applet 是否可以下载

所有动作都遵循标准 XML 语法。它们都有一个开始标签，其中包含有元素的名称和属性，一个可选的体（body）和一个结束标签。体和结束标签都可以为空，元素中只需有元素的名称和属性。例如：

```
<jsp:useBean id="user" class="com.mypackage.User">
    <jsp:setProperty name="first" value="John"/>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

```
        <jsp:setProperty name="last" value="Doe"/>
    </jsp:useBean>
```

jsp:useBean 动作有一个开始标签<jsp:useBean...>、一个体(包含有两个 jsp:setProperty 元素)和一个独立的结束标签</jsp:useBean>。

9.2.5 Eclipse 中的 JSP 的创建

下面介绍结合 Eclipse 和 Tomcat 技术开发一个简单 JSP 的例子，它将显示当前的日期，如能正确显示，则说明页面运行正常。具体步骤如下。

① 在 Eclipse 工作台的主菜单中选择“文件”→“新建”→“项目”。

② 此时将打开“新建项目”对话框，如图 9.11 所示，在对话框的左侧选择“Java”，然后在右侧选择“Tomcat Project”，然后单击“下一步”按钮。



图 9.11 “新建项目”对话框

③ 此时将打开“New Tomcat Project”对话框，在对话框中输入项目名，这里可以假设为 JSPProject，然后单击“下一步”按钮。

④ 现在将看到可以为 Web 项目的 Context 指定名字，并确认已选中“Can update server.xml file”选项，还可以指定一个子目录作为 Web 应用程序的根目录，假设保留默认值不变，如图 9.12 所示。

⑤ 单击“完成”按钮，现在在工作空间中创建了一个具有 WAR 结构的项目，如图 9.13 所示。

⑥ 选择所创建的项目，然后在其上单击鼠标右键。

⑦ 在上下文菜单中选择“文件”→“新建”。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 9.12　设置 Tomcat Web 应用程序设置

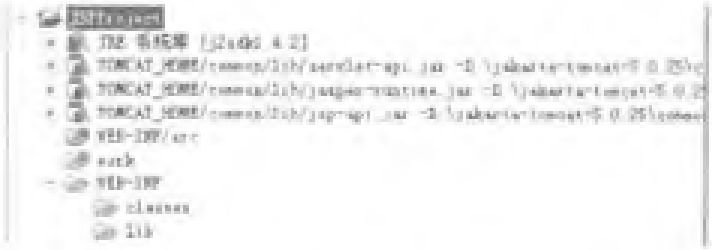


图 9.13　包资源管理器中所显示的 WAR 结构

⑧ 在“新建”文件窗口中指定文件名，假设为“index.jsp”。

⑨ 单击“完成”按钮，此时将创建 index.jsp 文件，并在编辑器中打开空白的 index.jsp 文件。

⑩ 将下列代码添加到 index.jsp 文件中，并保存这个文件。

```
<html>
    <body>
        <%java.util.Date d = new java.util.Date();%>
        Todays date is <%= d.getDate()%> and this jsp page worked!
    </body>
</html>
```

⑪ 单击工具条中的“Start Tomcat”按钮，启动 Tomcat 服务器。

⑫ 启动一个 Web 浏览器。在地址栏中输入 http://localhost:8080/JSPProject。此时会装入一个页面，将看到类似于图 9.14 所示的一条消息：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

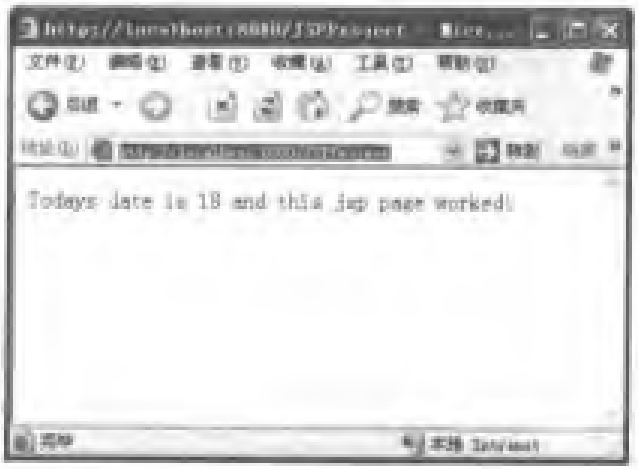


图 9.14　输入 http://localhost:8080/JSPProject 所显示的消息

因为截图时该 JSP 运行时间当时为 18 日，所以时间显示为 18 日。

9.3　Web 应用程序的打包和部署

Web 应用程序由一组 Servlet、JSP、HTML 页面和其他资源等组成。每种资源在 Web 应用项目中都有固定的存放目录。Web 应用程序可打包为 Web 归档文件（WAR 文件）。许多应用服务器都支持直接将 WAR 文件安装到服务器上。

9.3.1　部署描述符文件 web.xml

所有 WAR 文件包含一个 Web 应用部署描述符——web.xml。该文件用于描述 WAR 文件的内容并向应用服务器提供如何部署（安装）应用程序的信息。在 WAR 文件内，web.xml 应包含在 WEB-INF 文件夹中。该应用描述符文件通常由开发人员建立并由部署人员配置，从而将相关的应用程序安装到特定的应用服务器上。

web.xml 文件的一个功能是在 WAR 文件中指定可提供的或可用的 Web 组件。web.xml 文件中指定的每个 Web 组件需要指出提供组件运行时功能的资源文件。对于 Servlet，是 Java 类文件（.class）。

假设 Web 应用名为 HelloWorld，并发布到 Tomcat 应用服务器，则必须在 Tomcat 的安装目录下的/conf/server.xml 中的<Host>节中加入一行：

```
<Context path="/HelloWorld" docBase="myproject" debug="0"/>
```

一个 Context 对应于一个 Web 应用，每个 Web 应用是由一个或者多个 Servlet 组成的。当一个 Web App 被初始化的时候，它将用自己的 ClassLoader 对象载入部署描述符文件 web.xml 中定义的每个 Servlet 类。web.xml 文件有两部分：Servlet 类定义和 Servlet 映射定义。每个被载入的 Servlet 类都有一个名字，且被填入该 Context 的映射表中，和某种 URL 模式对应。当该 Context 被请求时，将查询映射表，找到被请求的 Servlet，并

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

执行。

下面以一个 web.xml 文件为例，介绍该文件的各个组成部分，并在文件中以注释的形式解释各个条目的含义。

```
<!-- 下面 web.xml 文件的第一行指定了 XML 的版本和字符编码。
```

第二行 DOCTYPE 指定文档类型，接下来声明了一个<web-app>元素，所有关于 Web 应用的配置元素都将加入到这个<web-app>元素中。

```
 -->
<?xml version="1.0" encoding="ISO-8859-1">
<!DOCTYPE web-app
PUBLIC "-/Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2.3.dtd">
<web-app>

  <!-- servlet 类定义部分
       <servlet-name>定义 servlet 的名字。
       <servlet-class>指定实现这个 servlet 的类。
      <init-parm>定义 servlet 的初始化参数，包括参数名和参数值。一个<servlet>元素中
      可以有多个<init-parm>，在 servlet 类中通过 getInitParameter(String name)方
      法访问初始化参数。
      当用户的 HTTP 请求无法匹配任何一个 servlet 的时候，default servlet 将被执行
-->
    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>
          org.apache.catalina.servlets.DefaultServlet
        </servlet-class>
        <init-param>
            <param-name>debug</param-name>
            <param-value>0</param-value>
        </init-param>
        <init-param>
            <param-name>listings</param-name>
            <param-value>true</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

  <!-- InvokerServlet
       处理一个 WEB APP 中的匿名 servlet。
       当一个 servlet 被编写并编译放入/WEB-INF/classes/中，却没有在/WEB-INF/web.xml
       中定义的时候，该 servlet 被调用，把匿名 servlet 映射成/servlet/ClassName 的形式。
       URL 模式映射：/servlet/*
  -->
    <servlet>
        <servlet-name>invoker</servlet-name>
        <servlet-class>
          org.apache.catalina.servlets.InvokerServlet
        </servlet-class>
        <init-param>
            <param-name>debug</param-name>
            <param-value>0</param-value>
        </init-param>
        <load-on-startup>2</load-on-startup>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    </servlet>

  <!-- JspServlet
       当请求的是一个 JSP 页面的时候（*.jsp）该 servlet 被调用
       它是一个 JSP 编译器，将请求的 JSP 页面编译成为 servlet 再执行
       URL 模式映射：*.jsp
   -->

    <servlet>
        <servlet-name>jsp</servlet-name>
        <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
        <init-param>
            <param-name>logVerbosityLevel</param-name>
            <param-value>WARNING</param-value>
        </init-param>
        <load-on-startup>3</load-on startup>
    </servlet>

    <!--  servlet 映射定义部分
          <servlet-mapping>元素用来设定客户端访问某个 servlet 的 URL.
          <servlet-mapping>使得程序中定义的 servlet 类名和客户访问的 URL 彼此独立。当
          servlet 类名发生改变，只要修改<servlet>元素中的<servlet-class>属性，而客户
          端访问的 URL 无须进行相应的改动。
  -->

    <!--  <servlet-name> 指定 servlet 的名字。这里的 servlet 名字应和<servlet>元素中
          定义的名字匹配。
          <url-pattern>指定访问这个 servlet 的 URL，这里只需给出相对于整个 Web 应用的相
          对 URL 路径
-->
    <servlet-mapping>
        <servlet-name>default</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>invoker</servlet-name>
        <url-pattern>/servlet/*</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>jsp</servlet-name>
        <url-pattern>*.jsp</url-pattern>
    </servlet-mapping>

</web-app>
```

9.3.2 Web 应用的部署

在 Tomcat 中部署 Web 应用系统有几种方式。最简单的一种方式是直接将 WAR 文件拷贝到 Tomcat 的安装目录下的 webapps 子目录中。

所部署的文件必须以“.war”作为扩展名。一旦 Tomcat 监听到这个文件，它将解开该文件包作为一个子目录，并以 WAR 文件的文件名作为子目录的名字。接下来，Tomcat 将在内存中建立一个 context，就好像在 server.xml 文件里建立一样。当然，其他必需的

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

内容，将从 server.xml 中的 DefaultContext 获得。

假设我们开发一个名为 HelloWorld 的 Web 应用系统，则将系统部署到 Tomcat 应用服务器，需要在 Tomcat 安装目录下的 webapps 目录中创建如下目录结构：

- /HelloWorld

Web 应用系统的根目录，所有的 JSP 和 HTML 文件都存放在此目录下。

- /HelloWorld/WEB-INF

Web 应用系统的部署描述符文件 web.xml 存放在此目录中。该目录包括 Web 应用相关的元数据。

- /HelloWorld/WEB-INF/classes

存放各种 class 文件，包括 Servlet 类文件在内。如果其中的 Servlet 和其他 Java 类以包的方式进行组织的话，那么 classes 下的目录结构应该和它们的包结构相同。

- /HelloWorld/WEB-INF/lib

存放 Web 应用程序所需的各种 JAR 文件。这些 JAR 文件将被部署到容器中，并有可能由类装载器装载。

在部署 Web 应用系统 HelloWorld 时，其步骤如下。

① 将 HTML 和 JSP 文件复制到/HelloWorld 目录中。

② 将 Servlet 程序编译后生成的 class 文件复制到/HelloWorld/WEB-INF/classes 目录中。需要注意的是，编译时需要将位于 Tomcat 安装目录下的/common/lib 目录下的 Java Servlet API 的 servlet-api.jar 文件设置为类路径。此外若 Servlet 属于某一包，如 package_name，则需要在/HelloWorld/WEB-INF/classes 目录中创建/package_name 子目录，并将该包中的 Servlet 的 class 文件都复制到该目录中，而不是直接复制到/HelloWorld/WEB-INF/classes 目录中。

③ 编写 web.xml 文件，并将该文件复制到/HelloWorld/WEB-INF 目录中。

④ 在 Tomcat 的安装目录下的/conf/server.xml 中的<Host>节中加入一行：

```
<Context path="/HelloWorld" docBase="myproject" debug="0"/>
```

9.4 使用 Lomboz 插件开发 J2EE 应用程序

Eclipse 的有一些插件可用来协助开发 J2EE 应用程序，如 Lomboz，Lomboz 不仅支持 JSP、Servlet，还支持 EJB 的开发，部署和测试等其他 J2EE 应用程序。Lomboz 将很多 Java 应用服务器、J2EE 组件和 Web 应用开发集成到 Eclipse 中，可以帮助 Java 开发者使用 Eclipse 建立、测试、部署 J2EE 应用。Lomboz 不但支持 JSP 等的语法高亮显示，而且还有代码辅助功能。

9.4.1 Lomboz 的获取

可从 http://www.objectlearn.com/网站下载 Lomboz 插件，目前最新的版本是 lomboz. 301。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

它的版本通常和 Eclipse 的版本有着对应关系。下载的 URL 为：http://forge.objectweb.org/project/showfiles.php?group_id=97 。

9.4.2 Lomboz 的安装

Lomboz 插件的安装和 Eclipse 中其他插件的安装一样，直接将下载的 Lomboz 插件的压缩文件解压缩到 Eclipse 安装目录下（即将 com.objectlearn.jdt.j2ee 文件夹和 com.objectlearn.jdt.j2ee.editors 文件夹解压缩到 eclipse 的 plugins 子目录下）。

9.4.3 Lomboz 在 Eclipse 中的配置

Eclipse 中 Lomboz 的配置步骤如下。

① 在 Eclipse 主菜单中选择“窗口”→“首选项”→“Java”→“类路径变量”，新建两个的类路径变量条目：

- 名称：JDK_TOOLS

路径：JDK 的 Tools.jar 的路径，如 C:/Program Files/j2sdk_nb/j2sdk1.4.2/lib/tools.jar

- 名称：TOMCAT

路径：Tomcat 安装目录，如：D:/jakarta-tomcat-5.0.25

② 在 Eclipse 工作台的主菜单中依次选择“窗口”→“定制透视图”→“窗口”→“显示视图”，并选中“Lomboz J2EE View”，如图 9.15 所示。

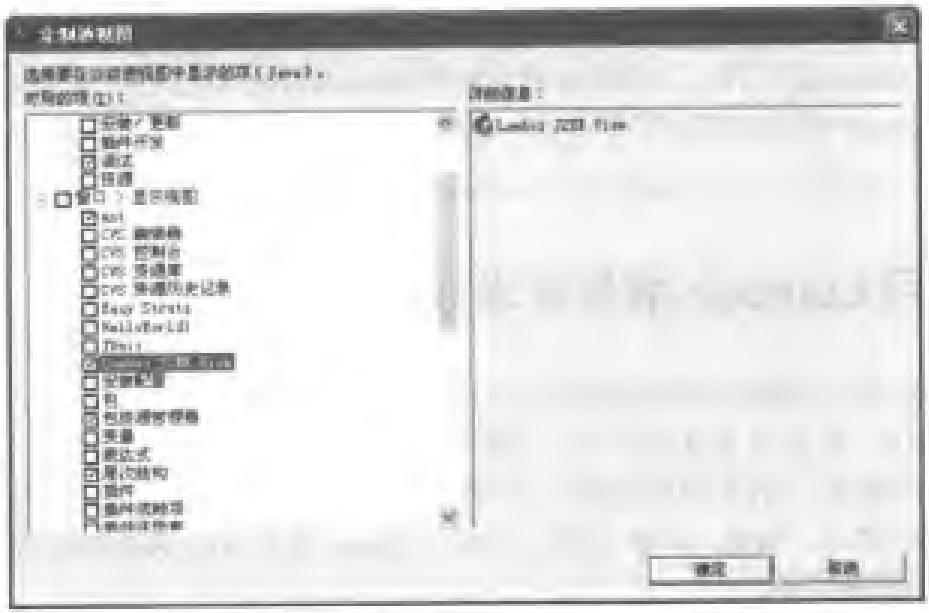


图 9.15 选中“Lomboz J2EE View”

③ 单击“确定”保存退出，然后后在 Eclipse 的工具栏中，将会添加一个按钮，如图 9.16 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 9.16　J2EE Project Outliner 按钮

④ 在 Eclipse 工作台的主菜单中选择“窗口”→“首选项”→“Java”→“新建项目”，将“作为 JRE 库使用”选项设置为“JRE_LIB 变量”。

⑤ 在 Eclipse 工作台的主菜单中选择“窗口”→“首选项”→“Lomboz”，设置 JDK Tools.jar 的包含完全路径的文件名，如 C:\Program Files\j2sdk_nb\j2sdk1.4.2\lib\tools.jar。

⑥ 单击“Lomboz”下的“Server Definitions”，如图 9.17 所示。



图 9.17　“Server Definitions”对话框

⑦ 在“Server types”下拉框中选择“Apache Tomcat v5.0.x”

在“Properties”选项卡下：

设置应用服务器的目录，如，D:\jakarta-tomcat-5.0.25

设置 Classpath Varible 名和变量，如分别为 TOMCAT_HOME 和 D:/jakarta- tomcat-5.0.25，单击“应用”按钮，再查看“Server ClassPath”选项卡，如果设置错误，Server Libraries 会显示错误标记。

⑧ 在“Server ClassPath”选项卡下，将显示服务器库文件，默认情况下 Tomcat 的库文件只包含 2 个 jar 文件：JDK 的 tools.jar 和服务器的 bootstrap.jar，单击“Add Library”按钮可添加库文件，单击“Add Variable”可添加路径变量。

9.4.4　Lomboz 的 JSP 应用程序开发示例

在 Eclipse 中配置好 Lomboz 后，即可使用它进行应用开发。下面将通过一个简单的

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

程序开发的示例，介绍基于 Lomboz 插件的 JSP 应用程序开发，其步骤如下。

① 在 Eclipse 工作台的主菜单中选择“文件”→“新建”→“项目”。

② 此时将打开“新建项目”对话框，在对话框左侧展开“Java”并选择“Lomboz J2EE Wizards”，然后在右侧选择“Lomboz J2EE Project”，如图 9.18 所示，然后单击“下一步”按钮。



图 9.18　新建 J2EE 项目

③ 在项目设置对话框中输入项目名，在本例中假设为 LombozJSP，然后单击“下一步”按钮。

④ 此时将打开“Java 设置”对话框，如图 9.19 所示，接受默认设置，单击“下一步”按钮。



图 9.19　Java 设置

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

⑤ 此时将打开“Create J2EE Module”对话框，如图 9.20 所示。



图 9.20 “Create J2EE Module”对话框

⑥ 在“Create J2EE Module”对话框中单击“Add”按钮，将弹出“Add Module”对话框，如图 9.21 所示



图 9.21 “Add Module”对话框

⑦ 在“Add Module”对话框中输入新增的模块名，假设为“dateShow”。

⑧ 在“Add Module”对话框中的“Targeted Servers”选项卡中，单击“Add”按钮，选择前面已配置好的服务器“Apache Tomcat v5.0.x”，如图 9.22 所示。

⑨ 然后单击“完成”按钮，将自动生成所创建的模块，此时在导航器视图中可看到项目中的文件结构如图 9.23 所示。

从图中可以看出，模块中包含自动生成的缺省 JSP 页面：error.jsp 和 index.jsp。

⑩ 单击 Eclipse 工具栏上的“J2EE Project Outliner 按钮”打开“Lomboz J2EE View”，如图 9.24 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

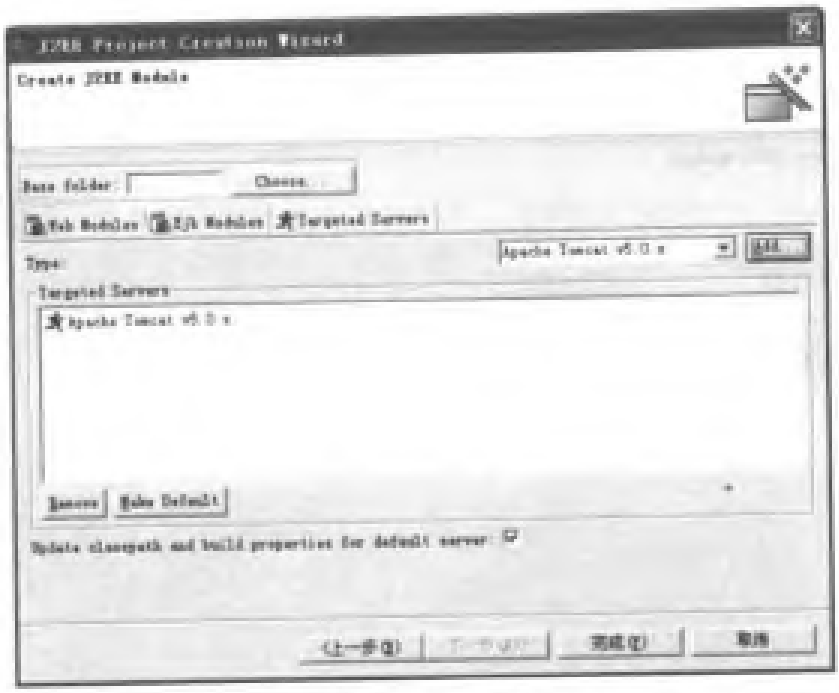


图 9.22　选择应用服务器

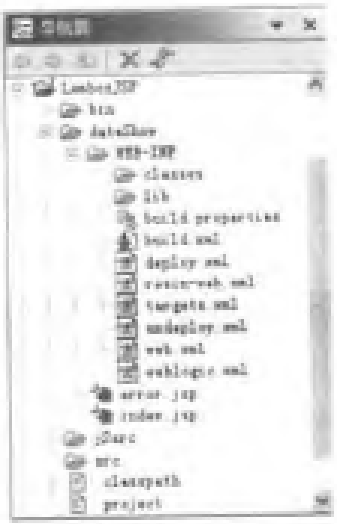


图 9.23　Lomboz 项目的文件结构

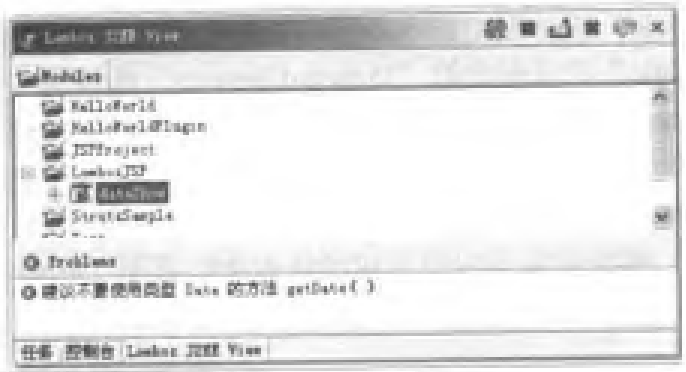


图 9.24　Lomboz J2EE View

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

⑪ 现在可以创建所需要的 JSP 文件。在本例中直接使用默认的 index.jsp，双击 LombozJSP 项目中的 dateShow 模块下的 WEB-INF 文件夹中的 index.jsp 文件，并对文件进行编辑，将它的内容改为：

```
<!-- Copyright (c) 2002 by ObjectLearn. All Rights Reserved. -->

<html>
    <head>
        <title>Welcome</title>
    </head>
    <body>
      <center>
        <%java.util.Date d = new java.util.Date();%>
      Todays date is <%= d.getDate()%> and this jsp page worked!
      </center>
    </body>
</html>
```

⑫ 在“Lomboz J2EE”视图中选中“dateShow”模块下的服务器“Apache Tomcat v5.0.x”，并在其上下文菜单中选择“Run Server”，如图 9.25 所示。



图 9.25　将程序在服务器上运行

⑬ 在浏览器中输入 http://localhost:8080/dateshow/index.jsp，显示如图 9.26 所示。

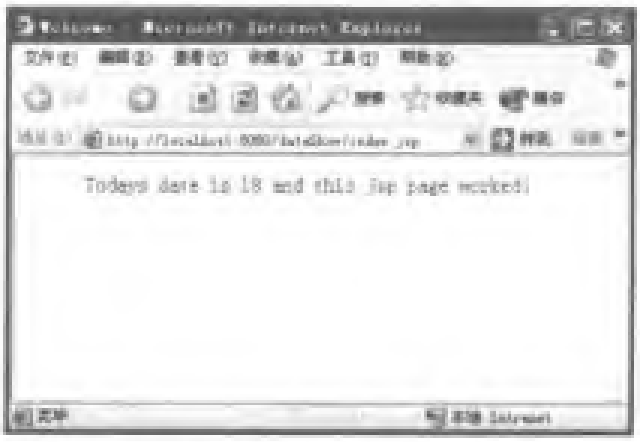


图 9.26　JSP 程序的运行结果

9.4.5　Lomboz 的 Servlet 应用程序开发示例

下面将通过一个简单计算 10 的阶乘的程序开发示例，介绍基于 Lomboz 插件的

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

Servlet 应用程序开发。在该示例中，创建项目和模块的步骤与创建 JSP 应用程序类似，其差别在于 Servlet 的创建，具体开发步骤如下。

① 在 Eclipse 工作台的主菜单中选择“文件”→“新建”→“项目”。

② 此时将打开“新建项目”对话框，在对话框左侧展开“Java”并选择“Lomboz J2EE Wizards”，然后在右侧选择“Lomboz J2EE Project”，并单击“下一步”按钮。

③ 在项目设置对话框中输入项目名，在本例中假设为 FactorProject，然后单击“下一步”按钮。

④ 此时将打开“Java 设置”对话框，接受默认设置，单击“下一步”按钮。

⑤ 此时将打开“Create J2EE Module”对话框，在“Create J2EE Module”对话框中单击“Add”按钮。

⑥ 此时将弹出“Add Module”对话框，在“Add Module”对话框中，输入新增的模块名，假设为“FactorModule”。

⑦ 在“Add Module”对话框中的“Targeted Servers”选项卡中，在类型下拉列表框中选择“Apache Tomcat v5.0.x”。

⑧ 单击“Add”按钮，将“Apache Tomcat v5.0.x”添加到“Targeted Servers”列表窗格中。

⑨ 然后单击“完成”按钮，将自动生成所创建的模块。

⑩ 在项目的上下文菜单中选择“新建”→“其他”，打开“新建”对话框窗口，如图 9.27 所示。



图 9.27 新建对话框

⑪ 在新建对话框窗口的左侧窗格中选中“Lomboz J2EE Wizards”，并在相应的右侧窗格中选中“Lomboz Servlet Wizard”。

⑫ 单击“下一步”按钮，将打开“Create a new Servlet”窗口，如图 9.28 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 9.28　“Create a new Servlet”对话框窗口

⑬　在对话框中的名称字段输入 Servlet 的名称，本例假设为“FactorServlet”，其他各项都接受默认值。

⑭　单击“下一步”按钮，将显示一个新的窗口，继续对 Servlet 进行设置，如图 9.29 所示。



图 9.29　“Create a new Servlet”对话框窗口的下一页

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

⑮ 在该设置页的“Web module”字段中直接输入前面创建的模块名“FactorModule”，也可单击“Browse”选择“FactorModule”。

在“Servlet name”字段中输入任一名称，通常取 Servlet 的类名，在本例中假设为“FactorServlet”。

在“Mapping URL”字段中输入 Servlet 对应的 URL，可为任意取值。主机名＋端口号＋模块名＋Mapping URL 将构成用户访问 Servlet 的完整的 URL。在本例中，假设“Mapping URL”取为“/Facotor”。注意，“/”一定要有。

⑯ 单击“完成”按钮，将在项目中将生成相应的 Servlet 文件 FactorServlet.java。

⑰ 现在可以对 FactorServlet.java 进行编辑了，将其他代码编写为如下内容并保存。

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServlet;

public class FactorServlet extends HttpServlet {

    public void service(ServletRequest request, ServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int n = 1;
        for (int i = 1; i <= 10; i++) {
            n *= i;
        }

        out.println("<html>");
        out.println("<head><title>TestServlet</title></head>");
        out.println("<body>");
        out.println("<P><center>10! = " + n + "</center></P>");
        out.println("</body></html>");
        out.close();

    }
}
```

⑱ 在“FactorModule”模块的上下文菜单中选择“Lomboz J2EE”→“Deploy Module”，将 Servlet 发布到 Tomcat 中。控制台视图中将会显示成功发布的信息，如图 9.29 所示。

⑲ 重新启动 Tomcat。

⑳ 在浏览器中输入 http://localhost:8080/FactorModule/Factor，显示如图 9.30 所示。

9.4.6 Lomboz 中的程序调试

JSP 程序的调试一直是 Java 开发工具的难点。对于 JSP 的调试，Lomboz 的原理是对要调试的 JSP 页面所产生的 Java 代码来进行调试。整个调试过程与 Java 应用程序的调试过程一样，这样不论是 JSP 本身的调试还是被 JSP 调用的对象的调试都被无缝地联

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

结到了一块。



图 9.30　发布结果在控制台中的输出



图 9.31　Servlet 程序的运行结果

Lomboz 支持调试语法、逻辑等各类错误。对于 JSP 程序中的语法错误，在编辑器视图的标记栏相应行的位置将会显示带白×的红色圆圈的错误标记，而警告信息则用带黑色惊叹号的黄色三角形表示，如图 9.32 所示。

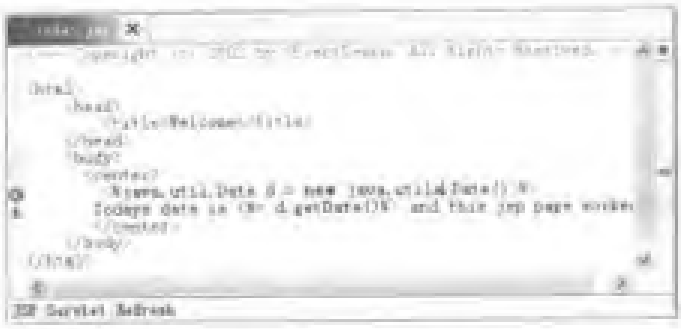


图 9.32　JSP 编辑器中的错误和警告标记

同时，错误和警告信息也将在任务视图和 Lomboz J2EE View 的“Problems”窗格中

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

显示。需要注意的是，首先需要在 Lomboz J2EE View 的“Modules”窗格中选择模块，错误和警告信息才会在“Problems”窗格中显示，并且都用带白×的红色圆圈的标记，如图 9.33（a）和 9.33（b）所示。



图 9.33（a） 任务视图中的错误和警告信息

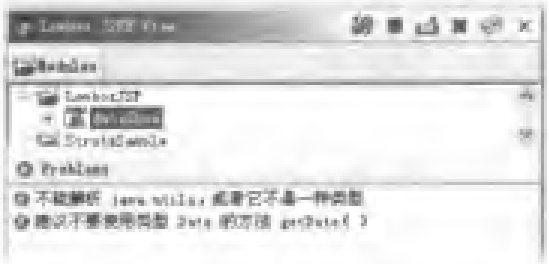


图 9.33（b） Lomboz J2EE View 的错误和警告信息

对于 JSP 中的逻辑错误，Lomboz 调试 JSP 的过程与普通 Java 应用程序的调试非常类似，因为它的原理就是先利用服务器生成 JSP 对应的 Java 文件，然后对他们进行调试。调试 JSP 的步骤如下。

① 打开 Lomboz J2EE View，选中相应的服务器 Apache Tomcat v5.0.x，单击鼠标右键，从上下文菜单中选择“Debug Server”，如图 9.34 所示。



图 9.34 调试服务器

② 如果服务器定义正确，Tomcat 将会启动。如果没有像预想的那样启动 Tomcat，那么请从这两个方面排错：Tomcat 环境变量和 Lomboz 的 server 文件。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 10 章　Eclipse 插件开发技术

插件是 Eclipse 平台功能的最小单元，可单独开发和提交。小的工具通常编写为单个插件，而一个复杂工具通常按功能划分为几个插件开发。除了平台运行时（runtime）的小内核外，所有的 Eclipse 平台功能都通过插件实现。例如，工作台用户接口插件完成界面的外观显示，资源管理插件完成维护或生成项目或文件资源等资源管理工作，而版本和配置管理插件则负责完成版本控制功能。因此，插件体系结构是 Eclipse 的主要特点之一。在这个体系中重要的概念是扩展点（extension points），也就是为插件提供的接口。每一个插件都是在现有的扩展点上开发，并可能还留有自己的扩展点，以便在这个插件上继续开发。因此，在现有插件上还可以再添加插件。

10.1　Eclipse 插件开发简介

面向对象的程序设计方法以及其概念和机制支持了软件的可重用性，插件技术是软件可重用性的基础。插件的本质是在不修改程序主体的情况下增强软件功能，通常一个插件就是一个新的功能。插件在系统已经存在的基础上，安装一个新功能时不需要更改整个软件的结构，而只需要将新功能文件安装到软件中就可以了，不会影响原有功能。

使用公开接口的插件，任何公司或个人都可以制作插件增强系统功能。插件实现了不同功能模块的无缝结合。也就是说，使用插件技术更易于实现一个庞大复杂的软件。插件接口的公开，使得更多人可以参与软件开发和维护，因此特别适合开放源代码软件项目的开发。

使用插件技术在程序中可以建立多个自定义的接口，使插件能够自由访问程序中的各种资源。插件易于安装，并且可以根据开发者（或用户）的需要进行个性化设置。

10.1.1　Eclipse 的插件结构

一个典型的插件由 JAR 库中的 Java 代码、一些只读文件及其他资源如图像、Web 模板、消息日志、本地代码库等组成。有一些插件根本就不包含代码，如以 HTML 形式提供给用户的在线帮助插件。一个单独的插件的代码库和只读内容共同存放在文件系统的一个目录中，或放在相同的 Web 服务器的目录下。

每个插件都有一个清单（manifest）文件表明该插件与其他插件的互连关系。插件通过添加到预定义的扩展点来向平台添加功能。一个插件的扩展点可被其他插件扩展。例如，工作台插件为用户首选项（Preference）声明一个扩展点。任一个插件都可通过定义

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

这个扩展点的扩展定义自己的用户首选项。

扩展点可能有一个对应的 API 接口。其他插件通过该扩展点的扩展实现这个接口。任何插件都可以定义新的扩展点及提供新的 API 以供其他插件使用。

Eclipse 启动时，平台运行时发现适用的插件，读取它们的清单文件，并建立一个内存中插件的注册。Eclipse 平台通过名字将扩展声明与它们对应的扩展点声明相匹配。其间出现的任何问题，诸如扩展丢失扩展点等，将被检测并记入日志。通过平台 API 可获取插件注册结果。Eclipse 启动后就不能再加入插件。

插件清单文件包含 XML。一个扩展点可声明其他特殊 XML 元素类型并在扩展中使用该元素类型。这使得提供扩展的插件与声明扩展点的插件间可传递任何信息。另外，不必激活起作用的插件或载入它们的任何代码就可从插件注册中得到清单信息。这个特性对于在大量安装的插件中只需使用一小部分插件的用户会话来说很重要。例如，声明用户首选项扩展点插件可以发现所有相关的用户首选项，并获取它们的显示名来构造首选项对话。这可以通过只使用注册中的信息而不用激活任何相关的插件来实现。

除非插件的代码被载入，否则插件几乎不消耗平台启动时间和内存开销。使用基于 XML 的插件清单使得编写支持插件创建的工具（如包含在 Eclipse SDK 中的插件开发环境）更容易。一个插件在它的代码确实需要运行时将被激活（activated）。一旦被激活，插件使用插件注册来发现和存取对它的扩展点起作用的扩展。例如，用户从列表中选择一个用户首选项时将激活相关的插件。这种方式激活插件不是自动发生，有一些 API 方法可用于激活插件。一旦激活，插件保持激活状态直到平台关闭。每个插件带有一个子目录来存储插件特定数据；这种机制允许插件可以储存运行间的重要状态。

平台运行时为应用声明一个特殊的扩展点。当平台的一个实例启动，应用名就通过命令行指定；初始只是声明哪个应用的插件被激活。因为预先可确定合适的插件集，并且无须激活任何插件就能在插件间交换大量信息，因此平台能为每个插件提供有关在哪个插件中操作的上下文相关信息。这个上下文在平台运行时不能交换，因此当上下文改变时不需要复杂的生命周期时间来告知插件。避免冗长的启动序列，因为不可预测的插件激活次序是常见的错误源。

Eclipse 平台由标准 Java 虚拟机的命令运行。每个插件分配给所属的 Java 类装载器，由 Java 类载入器负责装载插件的类（及 Java 资源包）。每个插件显式声明那些它希望直接存取类的插件的依赖。插件控制它的库中公共类和接口的可见性。该信息在插件清单文件中声明。可见性规则在运行时由插件类装载器执行。

Eclipse 平台的更新管理器下载并安装新的组件或更新已有组件的版本。更新管理器组建适合插件的新配置，在下次 Eclipse 平台启动时使用。如果更新或安装结果不满意，用户可以回退到先前的配置。

10.1.2 开发 Eclipse 插件的基本步骤

插件通过添加到预定义的扩展点来向平台添加功能。要将程序代码变成插件，需要：

（1）决定插件如何与平台集成

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在 Eclipse 平台中，用户开发的插件与基本插件没有任何不同。

（2）标识需要进行添加的扩展点以便与开发者的插件进行集成

（3）根据扩展点的规范来实现这些扩展

（4）提供清单文件 plugin.xml，它描述开发者正在提供的扩展以及代码的封装

（5）测试插件

（6）对插件进行封装

10.1.3 定义扩展点

如果希望其他插件扩展或定制插件的功能，则应定义扩展点。定义扩展点的第一步是将一些标记添加至插件的 plugin.xml 文件。下面是一个示例：

```
<?xml version="1.0" ?>
<plugin
   name="Simple Extension Point Plugin"
   id="org.eclipse.sample"
   version="1.0">
   <runtime>
      <library name="extpt.jar" />
   </runtime>
   <extension-point
      name="Sample Extension Point"
      id="sampleExtensionPoint"/>
</plugin>
```

这是定义开发者自己的扩展点所需的全部内容。连接至此扩展点的扩展结构不是由平台解释的，而是应遵循由扩展点提供程序定义的模式。惟一的限制是扩展标记必须是有效的 XML。

一旦定义了扩展点，就可以在运行时查询平台的插件注册表以查询和处理可能存在的任何扩展。在运行时，扩展显示为 IExtension 对象。扩展反过来由 IConfigurationElement 对象树组成，扩展标记中的每个元素为一个对象。以下代码片断查询与在上面定义的扩展点相关联的所有扩展的插件注册表。

```
   IPluginRegistry registry = Platform.getPluginRegistry();
   IExtensionPoint point =
registry.getExtensionPoint("org.eclipse.sample.sampleExtensionPoint");
   if (point != null) {
      IExtension[] extensions = point.getExtensions();
      System.out.println("Found " + extensions.length + " extensions");
   }
```

在一般情况下，扩展点与其扩展之间的惟一交互作用包含在扩展标记中。扩展可以直接在扩展标记的 XML 内提供一些数据或首选项设置，然后扩展点插件读取并处理该信息以提供一些定制行为。

开发者可能想要在运行时与扩展定义的对象交互，因为定义扩展点的插件通常不知道扩展中类和包的名称，所以在插件中必须定义接口来描述它期望的类结构。然后，可以使用 IExecutableExtension 来实例化扩展插件中的各个类，并在运行时使用该接口与对

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

象交互。

下面的 plugin.xml 文件的片段定义了 hook 至样本扩展点的扩展。

```
<extension
      point="org.eclipse.ui.views">
      <category
            name="HelloWorld"
            id="edu.sjtu.plugin.category1">
      </category>
</extension>
```

此扩展包括可执行扩展的标记。如前面所述，此示例中的 name 和 id 是任意的。对于此示例，可执行扩展的接口称为 ISomeInterface。

10.2 使用 JDT 开发 Eclipse 插件的示例

在本节中，我们将介绍使用 Java 开发环境开发插件，并在下一节中介绍如何直接使用插件开发环境开发插件。我们在此将开发一个“Hello，World”插件，该插件将实现自己的视图，并显示“Hello，World”。

在 Eclipse 中，包 org.eclipse.ui 及其子包中包含用来定义工作台用户界面（UI）API 的公共接口。这些接口中的许多接口都具有默认实现类，可以扩展这些类以便对系统提供简单的修改。在“Hello，World”示例中，我们将扩展工作台视图，以提供一个显示“Hello，World”进行问好的标号。

包 org.eclipse.ui 中的一个接口就是 IViewPart，它定义向工作台添加视图而必须实现的方法。类 ViewPart 提供了此接口的默认实现。视图部件负责创建显示视图所必需的小窗口。因此在插件程序中，需要 import 下列类：

```
org.eclipse.ui.part.ViewPart
org.eclipse.swt.widgets.Composite
org.eclipse.swt.widgets.Label
org.eclipse.swt.SWT
```

因此需要在插件项目的构建路径中添加 Eclipse 安装目录下的下列三个 JAR 文件：

```
\plugins\org.eclipse.core.runtime_2.1.1\runtime.jar
\plugins\org.eclipse.swt.win32_2.1.0\ws\win32\swt.jar
\plugins\org.eclipse.ui.workbench_2.1.0\workbench.jar
```

为了确保插件中的代码对 org.eclipse.ui 插件具有访问权，也可直接将相应的插件导入到项目中，其步骤如下。

① 从 Eclipse 工作台菜单中，选择“文件”→“导入”→“外部的插件和段”。

② 单击“下一步”按钮，此时将显示“导入插件和段”窗口。

③“导入插件和段”窗口中包含主机工作台实例中的所有插件的列表，选择 org.eclipse.ui。

④ 单击“添加必需的插件”按钮。

⑤ 单击“完成”按钮。“导航器”视图则应当包含 org.eclipse.ui 以及它需要的所有

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

插件（包括 org.eclipse.swt）。

在编写完插件代码并编译好项目后，如需将该插件添加到 Eclipse 工作台，需要通知平台我们将要添加视图。这一点是通过扩展 org.eclipse.ui.views 扩展点实现的。清单文件 plugin.xml 描述插件，包括插件的代码所在的位置以及正在添加的扩展，通过提供清单文件来注册扩展。

开发者可以使用任何 Java IDE 来构建 Eclipse 插件，但 Eclipse 本身就是一个很好的插件开发工具，并提供相应的 JDT 插件和 PDE 插件。

下面我们以“Hello，World”为例，作为插件实现，说明如何使用 JDT 创建平台插件，其步骤如下。

① 使用菜单栏选择“文件”→“新建”→“项目”，打开“新建项目”向导。

② 使用向导并以默认方式创建一个 Java 项目，取名为 HelloWorld。

③ 使用菜单栏选择“文件”→“新建”→“包”，以默认方式创建包，取名为 edu.sjtu.plugin。

④ 使用菜单栏选择“文件”→“新建”→“文件”，打开“新建文件”向导。

⑤ 以默认方式新建一个 Java 文件，取名为 HelloWorld.java。

⑥ 单击“完成”，创建该文件，并在文件中输入下列代码：

```
package edu.sjtu.plugin;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.SWT;
import org.eclipse.ui.part.ViewPart;
public class HelloWorld extends ViewPart {
    Label label;
    public HelloWorld() {
    }
    public void createPartControl (Composite parent) {
            label = new Label(parent, SWT.WRAP);
            label.setText ("Hello, World!");
    }
    public void setFocus() {
    }
}
```

⑦ 在包资源管理器窗格中，鼠标右击本项目，从上下文菜单中选择“属性”，指定项目的类路径，编辑配置程序的编译环境。

⑧ 在“属性”窗口的左面窗格选择“Java 构建路径”，然后在右面的窗格中选择“库”选项卡。

⑨ 在“库”选项卡下，单击“添加外部 JAR”按钮，添加 Eclipse 安装目录下的下列三个 JAR 文件：

```
\plugins\org.eclipse.core.runtime_2.1.1\runtime.jar
\plugins\org.eclipse.swt.win32_2.1.0\ws\win32\swt.jar
\plugins\org.eclipse.ui.workbench_2.1.0\workbench.jar
```

注意，随着 Eclipse 的版本不同，以上文件的路径名可能会有所不同。

⑩ 使用“新建文件”向导，以默认方式创建一个 XML 项目，文件名必须为 plugin.xml。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

⑪ 单击“完成”按钮，创建该文件，并在文件中输入下列代码：

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin
   id="edu.sjtu.plugin"
   name="HelloWorld"
   version="1.0"
   provider-name="Yuntao Zhang">

   <runtime>
      <library name="HelloWorld.jar"/>
   </runtime>
   <requires>
      <import plugin="org.eclipse.ui"/>
   </requires>

   <extension
         point="org.eclipse.ui.views">
      <category
            name="HelloWorld"
            id="edu.sjtu.plugin.category1">
      </category>
      <view
            name="HelloWorld1"
            category="edu.sjtu.plugin.category1"
            class="edu.sjtu.plugin.HelloWorld"
            id="edu.sjtu.plugin.view1">
      </view>
   </extension>

</plugin>
```

在 plugin.xml 中一共有四个主要的标签：plugin、requires、runtime、extension。其中 plugin 标签属性提供的是我们要开发插件的基本信息，除了 name、version、provider-name 等，最重要的是 id，它要求不能和现有的 Eclipse 插件标识符有冲突，因此本例中用包名作为插件的标识符。

runtime 标签指明的是开发的插件所在 JAR 包的文件名。由于我们打算将代码封装在 helloworld.jar 中，所以将其注册库名。

requires 标签中所列出的是需要的插件，需要用到 Eclipse 工作台和 SWT API，因此导入了 org.eclipse.ui 插件。

extension 标签是插件扩展点的信息，声明插件正向哪个扩展点进行添加。extension 中还包括 category 和 view 两个标签，要注意的是 category 和 view 标签的标识符的惟一性。

org.eclipse.ui.views 扩展具有几个不同的配置参数。首先为视图扩展声明类别。在工作台显示视图对话框中，可以使用类别来将相关视图集中在一起。我们定义了自己的类别“HelloWorld”，以便使它显示在它自己的组中。

org.eclipse.ui.views 是 Eclipse 系统提供的视图扩展点。本例是一个视图，这表明将在 org.eclipse.ui.views 扩展点上进一步开发，并且在 view 的属性中声明了插件的类名。

最后，为所创建的视图声明一个惟一的标识，并指定用来提供视图的实现类的名称。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

并为视图指定名称“HelloWorld1”，它将显示在“显示视图”对话框和视图的标题栏中。

Eclipse 为 plugin.xml 提供了默认的可视化编辑器，在编写 plugin.xml 过程中可以借助这个编辑器完成一些工作。

⑫ 选择 Eclipse 菜单栏“文件”→“导出”命令，将 HelloWorld 项目以 JAR 文件形式导出到 Eclipse 安装目录的 plugin 子目录下的某一目录中，同时将 plugin.xml 导出到该目录中。从而将插件安装到平台中，以便可以运行它。

⑬ 关闭并重新启动 Eclipse。Eclipse 启动时，将会搜索所有在插件路径下的插件并注册它们。

⑭ 在菜单栏中选择“窗口”→“显示视图”→“其他”，打开“显示视图”窗口。

⑮ 在“显示视图”窗口中，显示了在 plugin.xml 中 extension 的 category 标签中声明的 name 属性：HelloWorld，在 HelloWorld 结点中包含了 view（“视图”）标签的 name 属性：HelloWorld1，如图 10.1 所示。



图 10.1 “显示视图”窗口

⑯ 选中 HelloWorld1，并单击“确定”按钮，Eclipse 中将出现 HelloWorld1 窗口，如图 10.2 所示。



图 10.2 HelloWorld1 窗口

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

10.3 使用 PDE 开发插件

在 Eclipse 中使用常用的 Java 开发环境就可完成插件的开发，但在插件开发过程中的每次测试都要将代码打包、发布，然后再重新启动 Eclipse 系统。为了进一步便于进行插件开发，Eclipse 提供了进行插件开发的“插件开发环境（PDE）”，PDE 本身也是插件。

10.3.1 PDE 简介

插件开发环境 PDE 是一个工具，提供用来自动创建、处理、调试和部署插件的专门工具，使开发插件和扩展更方便、有效。在平台工作台中工作时，PDE 可以用来帮助开发者开发平台插件。PDE 提供了一组平台扩展添加项（视图、编辑器和透视图等），它们共同作用使得在工作台内开发插件的进程更为快速。

与一般平台工具一样，PDE 不是单独启动的工具。PDE 是作为包含在 SDK 中的一组插件来安装的。安装 PDE 时，工作台看起来是一样的，但是用户将看到工作台提供了新的透视图、向导、编辑器和视图。PDE 与平台融合在一起，并且通过新透视图来提供其功能。PDE 是基于平台和 JDT 的。

PDE 在平台中采用项目结构。当使用 PDE 时，将工作组织成项目、文件夹和文件。PDE 查看工作区中的项目，并注意将插件或段清单文件作为它们的直接子代的那些项目。如果一个项目具有插件清单，则 PDE 认为它是插件项目。

PDE 管理项目中的插件。每个项目都代表一个插件，并包含定义该插件的文件夹和文件，以及它将与平台交互作用的方式。这些插件将在用户的工作区中，并将显示在资源导航器及显示工作区资源的其他视图中。

PDE 中最重要的一个概念是主机和运行时工作台实例的概念。在使用 PDE 和其他工具来开发插件时，开发者正在运行的工作台实例称为主机实例。在此实例中可用的插件即为 Eclipse 工作台中所安装的插件。

在 PDE 中测试开发者所开发的插件时，可以启动另一个工作台实例，即运行时实例。此实例除了包含与主机实例相同的插件，还将包含开发者在主机实例中所开发的插件。PDE 启动器将正在开发的插件与主机插件合并，并创建运行时实例。

在运行时工作台实例中，将在 Eclipse 中安装的插件（即主机实例中的插件）称为外部插件。可以引用和浏览外部插件、查看外部插件的源代码以及调试它们，但不能修改外部插件，它们是只读的。运行时工作台实例中那些正在被开发的插件又称为工作区插件。针对工作区插件，不仅可以进行各种读操作，而且可以进行添加、删除、修改等写操作。PDE 将以不同的方式处理它们。

10.3.2 PDE 的配置

与其他插件一样，在 Eclipse 工作台中可以修改 PDE 的默认设置。在 Eclipse 工作台

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

主菜单中选择“窗口”→“首选项”打开“首选项”对话框，并在对话框的左侧窗格中选中“插件开发”类别，即在对话框的右侧窗格中显示插件开发首选项设置页面，如图 10.3 所示。

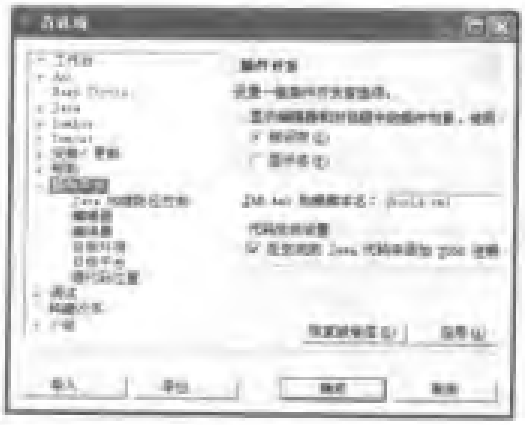


图 10.3 设置插件开发首选项

如图 10.3 所示，“插件开发”类别下包含“Java 构建路径控制”、“编辑器”、“编译器”、“目标环境”、“目标平台”和“源代码位置”条目，单击这些条目可进行相应的首选项设置。例如，在“Java 构建路径控制”条目中可以指定 PDE 在其中自动更新项目构建路径的区域，其默认值包括：

- 在创建新插件项目时；
- 在创建新段项目时；
- 在插件清单编辑器中修改依赖项时；
- 在 Java 至 PDE 项目转换期间。

10.3.3 “插件开发”透视图和相关视图

选择 Eclipse 菜单栏中的“窗口”→“打开透视图”，可打开“插件开发”透视图。插件开发透视图中的默认视图除了包括包资源管理器视图、编辑器、大纲视图、属性视图、任务视图、控制台视图外，还有一个“错误日志”视图，如图 10.4 所示。



图 10.4 “错误日志”视图

错误日志视图将捕获由平台或者开发者的代码所抛出的内部错误。这在运行时是很有用的，因为在开发期间可能会出现各种异常，触发错误时，会将它们添加到错误视图

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

中，可以通过双击它们来检查详细信息。可以在解决完错误后，清除该视图，并可以通过重新装入.log 文件来恢复视图的内容。条目是按逆时间顺序排列的（最晚的时间排在第一位），并且可以使用“属性”对话框来查看条目详细信息。

某些错误可能是分层的，即对于一个复杂问题可能有一个总条目，还有大量子条目，每个子条目详细列示个别问题。

除了错误日志视图，在插件开发透视图中还有一个很有用的视图——“插件注册表”视图。“插件注册表”视图不是插件开发透视图的默认视图。在运行时工作台实例中，选择“窗口”→“显示视图”→“其他”，并从“显示视图”对话框的“PDE 运行时”类别中选择“插件注册表”将打开“插件注册表”视图。

插件注册表视图显示它正在其中运行的实例的插件注册表的只读视图。借助插件注册表视图，开发者可以了解是否已启动开发者所开发的插件，以及平台能够识别哪些插件的扩展、扩展点、先决条件和运行时库等信息，如图 10.5 所示。

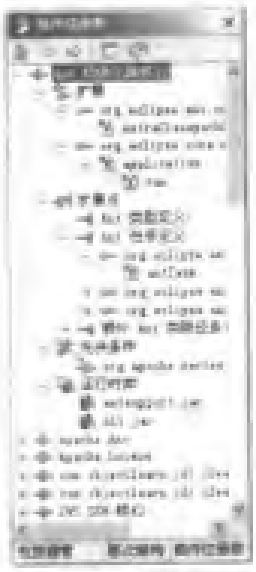


图 10.5 “插件注册表”视图

10.3.4 使用 PDE 开发插件的简单示例

下面，我们以开发一个简单的“Hello World”插件为例，介绍如何使用 PDE 开发插件，其开发步骤如下。

① 在菜单栏中选择“文件”→“新建”→“项目”，打开“新建项目”窗口。

② 在“新建项目”窗口中，在左边窗格中选中“插件开发”，并在右边的窗格中相应地选中“插件项目”，如图 10.6 所示。

③ 单击“下一步”按钮，在新窗口中，输入项目名 HelloWorldPlugin。PDE 的项目名称通常就是插件的标识符，因此需要保证项目名的惟一性，但是它们可以不相同。

④ 单击“下一步”按钮，显示“插件项目结构”窗口，采用默认选项，如图 10.7 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 10.6 创建插件项目



图 10.7 插件项目结构

⑤ 单击“下一步”按钮，显示“插件代码生成器”窗口，如图 10.8 所示。

在“插件代码生成器”窗口中，可以选择“创建空白插件项目”生成一个空插件，也可选择“使用代码生成向导创建插件项目”。为了说明插件结构，本例使用代码生成向导创建插件项目，并选择“Hello, World”代码生成向导。

⑥ 单击“下一步”按钮，进入“简单插件内容”窗口，采用默认设置，如图 10.9 所示。

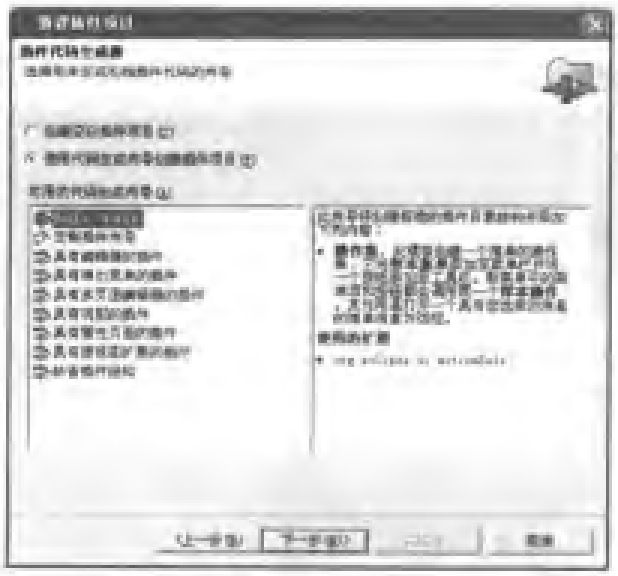


图 10.8 插件代码生成器



图 10.9 简单插件内容窗口

⑦ 单击“下一步”按钮，进入“样本操作集”窗口，采用默认设置，如图 10.10 所示。

“Hello, World”代码生成向导将创建一个简单的操作集，它将样本菜单添加至菜单栏，并将一个按钮添加至工具栏，新菜单中的菜单项和按钮都会调用同一个样本操作。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

其作用是打开一个具有所选择消息的简单消息对话框。



图 10.10　样本操作集

⑧ 单击“完成”按钮，完成插件项目的新建。因为所创建的项目与“插件开发”透视图相关联，因此将可自动转到“插件开发”透视图。

插件项目生成之后，项目中将会包含一个按默认方式生成的插件类 HelloWorld PluginPlugin.java 文件。打开该文件可以看到，该类继承了 AbstractUIPlugin 类，而 AbstractUIPlugin 类实现了 org.eclipse.ui.plugin 接口。

因为该项目是一个插件项目，所以单击 Eclipse 菜单栏中的“运行”按钮旁的下拉箭号，并选择“运行为”→“运行时工作台”。运行后，将生成一个和当前 Eclipse 几乎完全一致的平台，惟一的差别是在新的 Eclipse 平台上增加了一个“Hello，Eclipse world”插件（见图 10.11 中的工具条上的第 5 个按钮）。

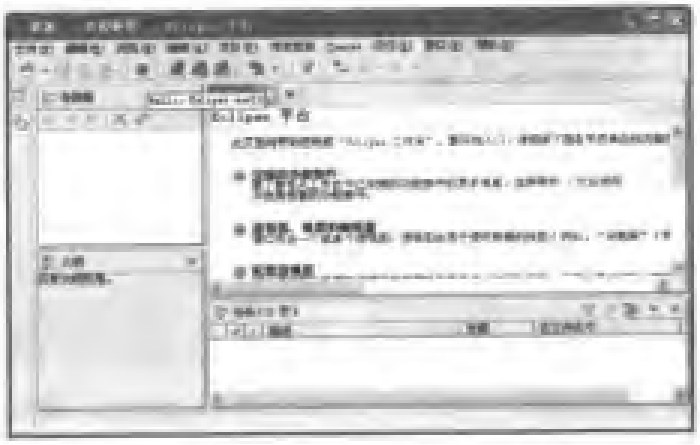


图 10.11　新生成的运行时工作台

与直接使用 Java 开发环境相比，使用运行时工作台方式调试插件，无须安装插件并启动运行时工作台，从而大大简化了插件开发的过程。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

10.3.5　插件清单编辑器

PDE 为插件清单文件 plugin.xml 提供了专门的插件清单编辑器。在刚才创建的项目中，双击插件清单文件 plugin.xml，插件清单编辑器即打开相应的插件清单文件，如图 10.12 所示。

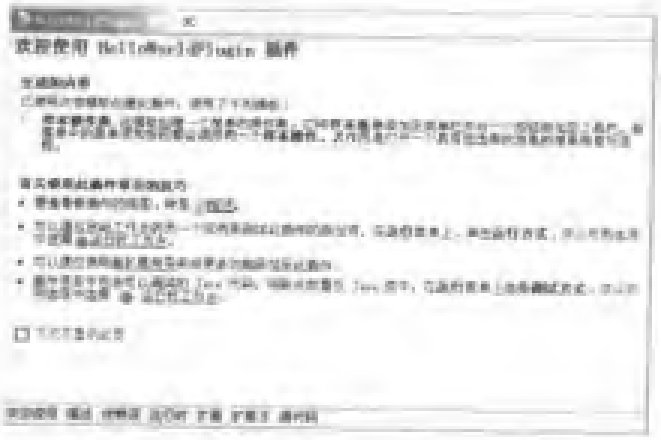


图 10.12　插件清单编辑器——欢迎使用

PDE 中的插件清单编辑器为多页编辑器，分别为欢迎使用、概述、依赖项、运行时、扩展、扩展点和源代码。

当使用模板向导创建插件时，清单编辑器将包含一个欢迎使用页面。如图 10.12 所示。欢迎使用用于总结由所选择的模板所生成的内容，并提供使用该插件项目的相关技巧。可以启用在欢迎使用页面左下角的“下次不显示此页”复选框，则在下次打开 plugin.xml 进行编辑时将不会出现该页面信息。

概述页面显示了插件的“警告和操作项”、“一般信息”、“扩展”、“运行时库”、“必需的插件”和“扩展点”，如图 10.13 所示。

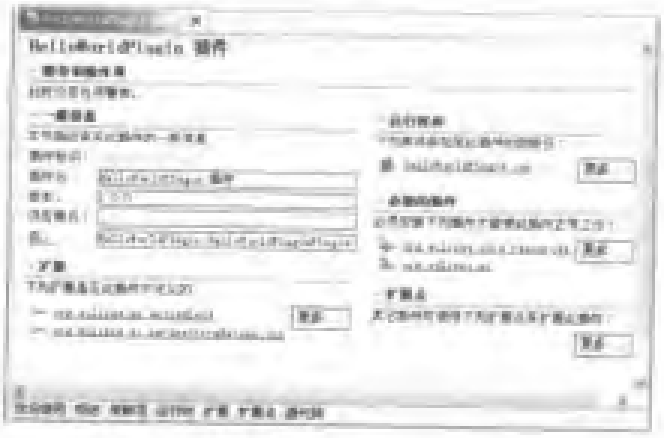


图 10.13　插件清单编辑器——概述

概述页面中的警告部分将显示警告问题以及未执行的与插件相关联的任务。例如，它将标记编译错误和警告，以及对其他插件的未解析的引用或者循环引用等。概述页面

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

显示了插件每个部分的最重要信息，并提供超级链接和“更多”按钮，以访问其他信息。在概述页面中可以编辑一些插件信息。此页上的其余信息只能够浏览。如果为插件指定了插件类，则超级链接将在 Java 编辑器中打开类。

依赖项页面显示该插件对其他插件的依赖性，如图 10.14 所示。



图 10.14　插件清单编辑器——依赖项

如图 10.14 所示，插件当前所依赖的插件显示在列表中。可以使用“添加”按钮添加更多的插件，或者使用上下文菜单从列表中除去插件。此列表中发生的更改将提示 PDE 更新插件清单并配置项目的构建路径，以便正在开发中的插件可以看见已导入的插件的类。

一旦将插件引用添加到列表中，就可以通过从弹出菜单中选择打开或者双击条目来浏览它。当在“必需的插件”列表中选择一个条目时，就可以在“导入详细信息”部分中重新导出依赖性。通过重新导出，可以使导入它的插件看见该插件。还可以更具体地考虑版本和需求的匹配。默认情况下，具有必需标识的任何插件都将可用。可以通过指定版本以及是否应该显式指定版本匹配来添加约束。

可以使用“依赖项状态”部分来获取依赖性图形信息。可以从下拉组合框中选择信息的类型。可以使用该部分来检查依赖性图形中的循环，引用所开发的插件的段以及需要该插件的其他插件。在各种情况下，结果都将显示在组合框下面的列表中。

运行时页面显示有关运行时库的信息，如图 10.15 所示。

当封装时，平台插件将提供它们在 JAR 库中的所有 Java 类。此页面定义源文件夹中的类在设计时是如何封装到库中的。在创建项目期间，向导已经创建了一个源文件夹和一个库。在此页面上，开发者可以定义多个源文件夹和库。

有关应该将其编译成库的源文件夹的信息实际上存储在单独的文件（build.properties）中。当更改此信息时，build.properties 也将更新。仅当未使用该文件时才可进行该操作。如果在使用该文件，则清单编辑器将指示正在使用该文件，开发者将只能浏览该信息。

默认情况下，库具有“代码”类型，因此假定库包含可执行代码和资源。访问这种库中的任何内容会导致激活插件及其所有依赖项。如果插件中的库不用来包含可执行代

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

码，则应将该库类型标记为“资源”。访问这种库中的“资源”不会激活插件，因此极大地提高了性能。

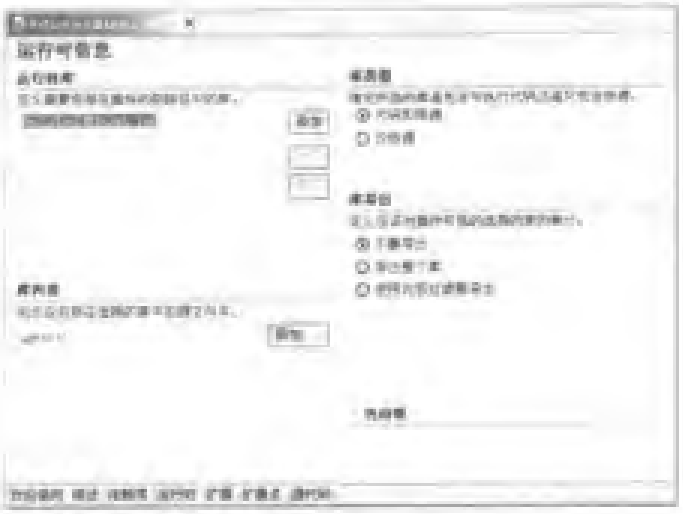


图 10.15 插件清单编辑器——运行时

还可以确定库的导出规则。默认情况下，其他人不能看到库中的任何类。如果不打算扩展插件，则这是可行的。如果要使其他插件也可以看到开发者所开发的类，则必须导出库。可以导出整个库，或者只导出与提供的内容过滤器相匹配的某些部分。

扩展页面可用来浏览和编辑插件扩展，如图 10.16 所示。



图 10.16 插件清单编辑器——扩展

扩展是用来为平台添加行为的中央机制。除非开发者的插件是可供其他插件使用的简单 Java API 库，否则，会将新行为作为扩展来添加。

插件为扩展点（由所需要的插件来定义）添加行为。扩展语法必须与扩展点定义相匹配，才能正确地进行处理。合格扩展点的总列表是由 PDE 通过遍历所有可视插件（工作区插件和外部插件）来计算的。

由 PDE 提供的一般扩展向导仅扫描扩展点。如果提供了扩展模式，则向导将从模式中抽取语法和扩展点元数据。除了一般向导之外，还可以获得大量扩展模板。这些是可

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

以用来生成插件项目初始内容的相同模板。如果它们在此处存在，则允许开发者创建扩展。可以使用简单视图来创建插件，稍后再添加编辑器或向导。

扩展点页面可用来浏览和编辑由插件定义的扩展点，如图 10.17 所示。

插件的扩展点可以供插件本身或者另一个插件来使用。定义扩展点时，必须提供标识和名称。标识应该是一个简单名称。其他插件将使用全限定版本，它们会将插件标识用做前缀。

除了显示扩展点的列表之外，此页面的右面窗格“扩展点使用情况”表示由其他插件添加到该扩展点的内容。开发者将可了解到谁真正扩展了扩展点。当开发者单击页面中的“添加”按钮时，将打开“新建扩展点”向导，在其中可定义新的扩展点。

在源代码页面中，开发者可以手工编辑 plugin.xml 文件。由于手工编辑该文件可能会导致在语法和语义上产生错误，因此，PDE 会在保存或切换选项卡时验证该文件以标记此类错误。

如果出现了语法错误，则 PDE 将锁定表单页面，直到修正了该错误为止。如果关闭存在错误的编辑器，然后再次打开它，则该编辑器将在源代码页面中打开，并且将不能访问其他页面，直到修正了源代码为止。

PDE 能够检测到的语义错误列示在“插件开发”→“编译器”首选项页面的“插件”选项卡中，如图 10.18 所示。



图 10.17　插件清单编辑器——扩展点



图 10.18　清单编译器的选项

可以将每一项的级别设置为“错误”、“警告”或“忽略”的其中之一。为了使 PDE 能够检测这些问题中的某些问题，开发者正在使用的扩展点必须具有与它们相关联的有效模式。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 11 章　Eclipse 的图形用户界面的构建

在 Eclipse 中涉及图形用户界面的编程往往是在较高层次进行的，开发者无须了解底层细节。例如，通过用户界面添加项的 plugin.xml 标记指定菜单和工具栏内容，而不需要进行任何 SWT 编程。另外，JFace 查看器和操作实现了应用程序与小窗口之间的普通交互作用。然而，了解底层 SWT 和 JFace 的相关知识对于了解平台的工作原理及开发图形用户界面程序仍然是有价值的。Eclipse 的联机帮助文档中包含了许多 SWT 和 JFace 的信息，本章的许多内容亦来源于此。如需更详细地了解相关内容，可进一步参见联机文档中的相关信息。

11.1　SWT 简介

标准小窗口工具箱简称 SWT，它是 Java 开发者的小窗口工具箱，它提供可移植的 API，并与计算机的底层操作系统的图形用户界面紧密集成。使用 SWT 小窗口可以直接实现工作台用户界面添加项，例如，视图、编辑器、向导和对话框等。

Java 抽象窗口工具箱 AWT 的类库内容极其丰富，共有 60 多个类和接口，包括了创建图形用户界面的所有工具。AWT 为较低级别的小窗口（例如列表、文本和按钮等）提供了平台集成的小窗口，但是没有提供对较高级别的平台组件，例如树或富文本等的访问权。由于 AWT 组件通过与平台相关的对等类实现，因此 AWT 与底层操作系统具有紧密的联系。为了保证程序的一致性，应用程序的开发人员经常不得不只能使用在所有平台上都可用的小窗口。

小窗口工具箱设计中的常见问题是可移植工具箱与平台集成之间的平衡。Swing 工具箱试图通过提供高级别小窗口，例如树、表和文本等的非本机实现来解决此问题。Swing 为基于图形用户界面的应用程序开发，提供了一套精美、丰富的基本组件以及一个能使得图形用户界面独立于特定平台的显示框架。

Swing 组件是用 100%纯 Java 实现的轻量级组件，没有本地代码，不依赖本机操作系统的支持，这是它与 AWT 组件的最大区别。

Swing 工具箱提供了许多功能，但在 Swing 中开发的应用程序的外观总是和同一操作系统平台下的其他软件显得差异很大。为了解决这一问题，通常使用平台外观模拟层帮助应用程序看起来更像平台中的其他程序，但是与用户交互方式仍然有很大的不同。这使得基于 Swing 的图形用户界面的程序的外观总比为特定操作系统平台专门开发的应用程序的外观稍逊一筹。

SWT 通过定义在所有受支持的平台上提供的常见可移植 API 来解决此问题，并尽可

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

能使用本机小窗口在每个平台上实现该 API。这使得工具箱在所有平台上维护一致的编程模型时立即反映底层操作系统的图形用户界面外观中的任何更改。

SWT 最大化了操作系统的图形组件 API，就是说只要操作系统提供了相应图形的组件，那么 SWT 只是简单应用 JNI 技术调用它们，只有那些操作系统中不提供的组件，SWT 才自己去做一个模拟的实现。可以看出 SWT 性能上的稳定大多时候取决于相应操作系统图形组件的稳定性。

SWT 是比 Swing 更紧密地映射到底层操作系统的本机图形技术，这不仅使得 SWT 更快速，而且使得 Java 程序具有更像本机应用程序的外观和感觉。使用这个新的图形用户界面 API 可能会限制 Eclipse 工作台的可移植性，不过 Eclipse 对 SWT 的使用只会影响 Eclipse 自身的可移植性，而使用 Eclipse 构建的任何 Java 应用程序都不会受到影响，除非它们使用 SWT 而不是使用 Swing 或 AWT。

SWT 包含许多功能部件，但在开发图形用户界面程序中主要涉及小窗口、布局和事件。

11.2 SWT API 的结构

SWT 本身仅仅是 Eclipse 组织为了开发 Eclipse IDE 环境所编写的一组底层图形界面 API。SWT 在 Eclipse 开发环境中作为一个插件形式存在，可以在 Eclipse 安装路径下的 \plugins 路径下的众多子目录下去搜索 swt.jar 文件，具体的目录和 Eclipse 的版本有关，例如目录\plugins\org.eclipse.swt.win32_2.1.0\ws\win32\swt.jar。在找到的 JAR 文件中包含了 SWT 全部的 Java 类文件。

在 Eclipse 中，进行图形用户界面的编程，通常需要用到 org.eclipse.swt.widgets、org.eclipse.swt.events 和 org.eclipse.swt.layout 这三个最主要的包。

所有的 SWT 类都用 org.eclipse.swt 作为包的前缀。org.eclipse.swt.widgets 包中是常用的、核心 SWT 窗口小部件的公有 API 类定义所在。最常用的图形组件基本都被包括在 org.eclipse.swt.widgets 包中，比如按钮（Button）、文本框（Text）和表格（Table）等。其中两个最重要的组件是 Shell 和 Composite。Shell 相当于应用程序的主窗口框架。Composite 相当于 SWING 中的 Panel 对象，充当着组件容器的角色。当需要在窗口中加入一些组件时，最好使用 Composite 作为其他组件的容器，然后再去 org.eclipse.swt.layout 包中找出一种合适的布局方式。

org.eclipse.swt 定义了 SWT 需要使用的常量和异常的类。这个包由三个类组成：SWT, SWTException 和 SWTError。SWT 类包含了 SWT 库所需要的常量如键盘、错误、颜色、布局、文本样式、按钮等常量。

SWT 对用户操作的响应，比如鼠标或键盘事件，也是采用了 AWT 和 Swing 中的 Observer 模式。org.eclipse.swt.events 定义了 SWT 组件使用的 typed events、listeners 和 events。这个包有三组不同的类：Listener 接口，Adapter 类和 Event 类。org.eclipse.swt.events 包中提供了对 SWT 事件监视器（Event Listener）的支持，如 Button 的 SelectionListener，Mouse 的 MouseListener、MouseMoveListener 和 MouseTrackListener 等，还有与这些对

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

应 Listener 的默认 Adapter 实现类和 Event 类。

SWT 对组件的布局也采用了 Swing 或 AWT 中布局（Layout）和布局结构（Layout Data）结合的方式。org.eclipse.swt.layout 包是控制图形用户界面程序布局的类所在，如 FillLayout、GridLayout、RowLayout 和 FormLayout。

在 org.eclipse.swt.custom 包中，包含了一些可自定义的窗口小部件，即对一些基本图形组件的扩展，比如其中的 CLabel，就是对标准 Label 组件的扩展，上面可以同时加入文字和图片，也可以加边框。StyledText 是 Text 组件的扩展，它提供了丰富的文本功能，比如对某段文字的背景色、前景色或字体的设置。这些对于开发复杂的图形用户界面程序，非常有帮助。

org.eclipse.swt.graphics 包中包含了 SWT 中 graphic 类，可以找到针对图片、光标、字体或绘图的 API。比如可通过 Image 类调用系统中不同类型的图片文件。通过 GC 类实现对图片、组件或显示器的绘图功能。

org.eclipse.swt.dnd 提供了对拖放的支持。org.eclipse.swt.printer 提供了对打印服务的支持。org.eclipse.swt.print 包含了对 SWT 窗口组件提供打印支持的类。

对不同平台，Eclipse 还开发了一些富有针对性的 API。例如，org.eclipse.swt.ole.win32 提供了 SWT 中 Win32 OLE 实现的一些类。在 Windows 平台，可以通过 org.eclipse.swt.ole.win32 包很容易地调用 OLE 控件，这使 Java 程序内嵌 IE 浏览器或 Word、Excel 等程序成为可能。

11.3　小窗口

当使用平台工作台扩展来添加用户界面元素时，用于启动 SWT 的机制是由工作台来处理的。

11.3.1　小窗口应用程序的结构

典型的独立 SWT 应用程序具有下列结构。

① 创建显示，它表示 SWT 会话。

② 创建一个或多个 Shell，它充当应用程序的主窗口。

③ 创建 Shell 内部需要的任何其他小窗口。

④ 初始化小窗口的大小和其他必需的状态。为需要处理的小窗口事件注册侦听器。

⑤ 打开 Shell 窗口。

⑥ 运行事件调度循环，直到发生退出情况为止（通常是在用户关闭主 Shell 窗口的情况下）。

⑦ 除掉显示。

显示表示 SWT 与底层平台的图形用户界面系统之间的连接。显示主要用来管理平台事件循环和控制用户界面线程与其他线程之间的通信。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

对于大多数应用程序，在创建任何窗口之前必须创建显示，并且当关闭 Shell 时，必须除掉显示。

Shell 是由操作系统平台窗口管理器管理的一个“窗口”。顶级 Shell 是作为显示的子代创建的那些 Shell。这些窗口是在用户使用应用程序时移动、调整大小、最小化和最大化的窗口。辅助 Shell 就是作为另一个 Shell 的子代创建的那些 Shell。这些窗口通常用做只存在于另一个窗口的上下文中的对话框窗口或其他瞬时窗口。

不是顶级 Shell 的所有小窗口都必须具有父代。顶级 Shell 没有父代，但是，它们是通过与特定显示相关联而创建的。可以使用 getDisplay()来访问此显示。所有其他小窗口都是作为顶级 Shell 的子代（直接或间接）来创建的。组合体小窗口是可以具有子代的小窗口。

当看到应用程序窗口时，可以将它看做小窗口树，或者看做层次结构，其根目录就是 Shell。根据应用程序的复杂性，应用程序窗口可以是具有 Shell 的单个子代、几个子代或者具有子代的组合体的嵌套层。

以下代码片段是小窗口应用程序的一个示例，通过注释的形式说明了它的结构。它是 Eclipse 中 org.eclipse.swt.examples.helloworld.HelloWorld2 应用程序的一部分。由于该应用程序只显示字符串“Hello World”，因此不需要向任何小窗口事件注册。

```
public static void main (String [] args) {
  Display display = new Display ();  //创建一个显示
  Shell shell = new Shell (display);  //创建一个 shell
  Label label = new Label (shell, SWT.CENTER); //创建 label 小窗口
  label.setText ("Hello_world"); //初始化 label 小窗口
  label.setBounds (shell.getClientArea ()); //初始化 label 小窗口
  shell.open (); //打开 shell 窗口
  while (!shell.isDisposed ()) { //运行事件调度循环
    if (!display.readAndDispatch ()) display.sleep ();
  }
  display.dispose (); //除掉显示
}
```

11.3.2 小窗口应用程序的系统资源管理

当应用程序创建小窗口时，SWT 将立即创建底层平台小窗口。因此，不再需要根据底层操作小窗口是否存在而执行不同操作的代码。SWT 还允许将小窗口的大部分数据保存在平台层而不是在工具箱中复制。这就意味着工具箱的小窗口生命周期的概念必须符合底层图形用户界面系统的规则。

创建小窗口时，大多数图形用户界面平台都要求开发者指定父代。由于只要开发者创建了工具箱小窗口，SWT 就会创建平台小窗口，因此，必须在构造函数中为小窗口指定父代小窗口。

在一个图形化的操作系统中开发程序，需要调用系统中的资源，如图片、字体、颜色等。通常这些资源都是有限的，开发人员需要非常小心地使用这些资源：当不再使用它们时，需要尽快释放，不然将会耗尽系统资源。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

SWT 是用 Java 开发的，Java 语言本身的一大优势就是 Java 虚拟机的“垃圾回收机制”，程序员通常不用处理变量的释放和内存的回收等问题。然而，SWT 并没采用 JVM 的垃圾回收机制去处理操作系统的资源回收问题，SWT 要求开发人员显式地释放已经分配的任何操作系统资源。主要原因是 Java 虚拟机的垃圾回收机制是不可控的，也就是说开发人员不能知道，也不可能做到在某一时刻让 Java 虚拟机回收资源。因此在 SWT 中经验法则是：“谁创建，谁释放”。例如，如果需要创建 Font：

```
Font font = new Font (display, "Courier", 10, SWT.NORMAL);
```

那么就应该在不在需要这个 Font 的时候调用：

```
font.dispose();
```

如果程序调用某一组件的 dispose()方法，那么所有这个组件的子组件也会被自动调用 dispose()方法而销毁。通常这里指的子组件与父组件的关系是在调用组件的构造函数时形成的，例如：

```
Shell shell = new Shell();
Composite parent = new Composite(shell,SWT.NULL);
Composite child = new Composite(parent,SWT.NULL);
```

以下进一步说明此法则的特定规则：

- 当使用构造函数来创建小窗口或图形对象时，使用完后必须显示释放它。
- 如果没有使用构造函数就获取小窗口或图形对象，因为未分配它，因此不应显式释放它。
- 如果将对小窗口或图形对象的引用传送至另一个对象，则在仍使用它时，注意不能释放它。
- 当用户关闭 Shell 时，将递归地销毁 Shell 及其所有子代小窗口。在此情况下，不需要除掉小窗口本身，因为父组件被销毁，子组件也同时被销毁。然而，必须释放与那些小窗口一起分配的所有图形资源。
- 如果创建图形对象以便在其中一个小窗口的生命周期内使用它，则在除掉小窗口时必须除掉图形对象。这可以通过向小窗口注册销毁侦听器，并在接收到销毁事件时释放图形对象来实现。

这些规则有一个例外。简单的数据对象，例如矩形和点，不使用操作系统资源。它们没有 dispose()方法，因此也不需要释放它们。

11.3.3 控件

在 SWT 类层次结构中，小窗口是可以放置在另一个小窗口内的任何用户界面对象的抽象类。控件是通常在底层平台中具有相应表示法（由操作系统窗口句柄指示）的小窗口。

org.eclipse.swt.widgets 包定义了 SWT 中的一组核心小窗口。表 11.1 总结了在此包中提供的控件的具体类型及其用途。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

表 11.1 org.eclipse.swt.widgets 包中提供的控件的具体类型及用途

小窗口	用 途	样 式	事 件
按钮	可选择的控件，当按下和/或释放该控件时就会发出通知	BORDER, CHECK, PUSH, RADIO, TOGGLE, FLAT, LEFT, RIGHT, CENTER 和 ARROW（带 UP 和 DOWN）	销毁，控件*和选择
画布	组合体控件，提供用于绘制任何图形的面板，通常用来实现定制控件	BORDER, H_SCROLL, V_SCROLL, NO_BACKGROUND, NO_FOCUS, NO_MERGE_PAINTS, NO_REDRAW_RESIZE 和 NO_RADIO_GROUP	销毁和控件*
插入符	通常出现文本的插入点		销毁
组合框	可选择的控件，允许用户从字符串列表中选择字符串，或者有选择性地将新值输入到可编辑的文本字段中，通常在有限的空间需要可用的字符串的下拉显示而不是使用单个选择列表框时使用它	BORDER, DROP_DOWN, READ_ONLY 和 SIMPLE	销毁，控件*，默认选择，修改和选择
组合体	能够包含其他小窗口的控件	BORDER, H_SCROLL 和 V_SCROLL	销毁和控件*
CoolBar	组合体控件，允许用户动态地重新定位栏中包含的 cool 项	BORDER	销毁和控件*
CoolItem	可选择的用户界面对象，表示 cool 栏的动态可定位区域	DROP_DOWN	销毁
组	组合体控件，它将其他小窗口分组，并利用边界和/或标签来包围它们	BORDER, SHADOW_ETCHED_IN, SHADOW_ETCHED_OUT, SHADOW_IN, SHADOW_OUT 和 SHADOW_NONE	销毁和控件*
标号	不可选择的控件，它显示字符串或图像	BORDER, CENTER, LEFT, RIGHT, WRAP, SEPARATOR (with HORIZONTAL,SHADOW_IN, SHADOW_OUT, SHADOW_NONE, VERTICAL)	销毁和控件*
列表	可选择的控件，允许用户从字符串列表中选择一个或多个字符串	BORDER, H_SCROLL, V_SCROLL,SINGLE 和 MULTI	销毁，控件*，选择和缺省选择
菜单	包含菜单项的用户界面对象	BAR, DROP_DOWN, NO_RADIO_GROUP 和 POP_UP	销毁，帮助，隐藏和显示
菜单项	表示菜单中项的可选择的用户界面对象	CHECK, CASCADE, PUSH, RADIO 和 SEPARATOR	销毁，配备，帮助和选择
进度条	不可选择的控件，它为用户显示进度，通常采用条形图形的形式	BORDER, INDETERMINATE, SMOOTH, HORIZONTAL 和 VERTICAL	销毁和控件*
窗框	可选择的控件，允许用户在父代窗口中拖动窗框的外边框，用来允许用户通过重新定位子代小窗口的分界线来调整它们的大小	BORDER, HORIZONTAL 和 VERTICAL	销毁，控件*和选择
范围	可选择的控件，它表示数值的范围	BORDER, HORIZONTAL 和 VERTICAL	销毁，控件*和选择

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

续表

小窗口	用 途	样 式	事 件
滚动条	可选择的控件，它表示正数值的范围，用于具有 V_SCROLL 和（或 H_SCROLL 样式的组合体	HORIZONTAL 和 VERTICAL	销毁和选择
Shell	由 OS 窗口管理器管理的窗口。Shell 可以是"显示"（顶级 Shell）或者另一个 Shell（辅助 Shell）的父代	BORDER、H_SCROLL、V_SCROLL、CLOSE、MIN、MAX、NO_TRIM、RESIZE 和 TITLE（另见 SHELL_TRIM 和 DIALOG_TRIM）	销毁、控件*、激活、关闭、释放、取消图标化和图标化
滑块	可选择的控件，它表示数值的范围。滑块不同于标尺，它提供了一个可拖动的手势，可以根据范围调整当前值	BORDER、HORIZONTAL 和 VERTICAL	销毁、控件* 和选择
TabFolder	组合体控件，它将集中可以由用户使用加了标签的标记来选择的页面	BORDER	销毁、控件* 和选择
标记项	与标记文件夹中页面的标记相对应的可选择用户界面对象		销毁
表	可选择的控件，它显示用户可以选择的表项列表。项是按行提供的，它显示表示项的不同方面的多列	BORDER、H_SCROLL、V_SCROLL、SINGLE、MULTI、CHECK、FULL_SELECTION 和 HIDE_SELECTION	销毁、控件*、选择和默认选择
表列	用来表示表中列的可选择用户界面对象	LEFT、RIGHT 和 CENTER	销毁、移动、调整大小和选择
表项	用来表示表中的项的可选择用户界面对象		销毁
文本	可编辑的控件，允许用户向其中输入文本	BORDER、SINGLE、READ_ONLY、LEFT、CENTER、RIGHT、WRAP 和 MULTI（带 H_SCROLL 和 V_SCROLL）	销毁、控件*、默认选择、修改和验证
工具栏	组合体控件，它支持可选择的工具栏项的布局	BORDER、FLAT、WRAP、RIGHT、SHADOW_OUT HORIZONTAL 和 VERTICAL	销毁和控件
工具项	用来表示工具栏中项的可选择用户界面对象	PUSH、CHECK、RADIO、SEPARATOR 和 DROP_DOWN	销毁和选择
跟踪器	用来实现橡皮条状矩形的用户界面对象	LEFT、RIGHT、UP、DOWN、RESIZE	销毁、移动和调整大小
树	可选择的控件，它显示用户可以选择的树项的分层列表	BORDER、H_SCROLL、V_SCROLL、SINGLE、MULTI 和 CHECK	销毁、控件*、选择、默认选择、折叠和展开
树项	可选择的用户界面对象，表示树中的树项的层次结构		销毁

控件*：指从控件中继承的事件，焦点进入、焦点离开、帮助、键按下、键抬起、鼠标双击、鼠标按下、鼠标进入、鼠标离开、鼠标悬浮、鼠标轨迹、鼠标移动、移动、绘制和调整大小。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

11.3.4 定制小窗口

有时，SWT 中提供的控件可能没有任何一个能满足应用程序的需要。在这些情况下，你可能想通过自己的定制小窗口来扩展 SWT。SWT 本身提供了一个包 org.eclipse.swt.custom，它不包含在 SWT 控件的核心组中，但作为实现平台工作台所需要的定制控件。表 11.2 总结了在此包中提供的控件的具体类型及其用途。

表 11.2 org.eclipse.swt.custom 包中提供的控件的具体类型及其用途

控 件	用 途	样 式	事 件
CCombo	类似于组合框，但是定制绘制以便允许使用没有边界的组合框。开发此类的目的是为了在表单元内使用组合框	BORDER, FLAT 和 READ_ONLY	销毁、控件* 和选择
CLabel	类似于标签，但是支持剪辑带有省略的文本。当在活动工作台视图中查看时，还支持背景颜色的倾斜效果，不支持覆盖	CENTER, LEFT, RIGHT, SHADOW_IN, SHADOW_OUT 和 SHADOW_NONE	销毁和控件*
CTabFolder	类似于 TabFolder，但是它支持标记（顶部或底部）和边界的可按外观的附加配置	BORDER, FLAT, BOTTOM 和 TOP	销毁、控件* 和选择
CTabItem	与 CTabFolder 中的页面的选项卡相对应的可选择用户界面对象		销毁和控件*
SashForm	组合体控件，将它的子代布局为按行或列排列，并使用“窗框”来分隔它们，以使用户可以调整它们的大小	BORDER, HORIZONTAL 和 VERTICAL	销毁和控件*
ScrolledComposite	组合体控件，滚动它的内容，并可选择伸展其内容以填充可用空间	BORDER, H_SCROLL 和 V_SCROLL	销毁和控件*
StyledText	可编辑的控件，允许用户输入文本，控件内的文本范围可以具有截然不同的字体和颜色（前台和后台）	BORDER, FULL_SELECTION, MULTI, SINGLE, WRAP 和 READ_ONLY	销毁、控件*、ExtendedModify、LineGetBackground、LineGetSegments、LineGetStyle、修改、选择、验证和 VerifyKey
TableTree	可选择的控件，它显示用户可以选择的项的分层列表，项是按行提供的，它显示表示项的不同方面的多列	BORDER, SINGLE, MULTI, CHECK 和 FULL_SELECTION	销毁、控件*、选择、缺省选择、折叠和展开

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

续表

控 件	用 途	样 式	事 件
TableTreeItem	可选择的用户界面对象，表示 TableTree 中的树项的层次结构		
ViewForm	组合体控件，它在水平方向布局三个子代，并允许有计划地控制布局和边界参数，用于在工作台中来实现视图的标号/工具栏/菜单局部栏	BORDER 和 FLAT	销毁和控件*

控件* 指从控件中继承的事件：焦点进入、焦点离开、帮助、键按下、键抬起、鼠标双击、鼠标按下、鼠标进入、鼠标离开、鼠标悬浮、鼠标抬起、鼠标移动、移动、绘制和调整大小。

在实现定制小窗口之前，应当考虑以下几个重要问题。

- 现有 SWT 控件可以用做更简单或更普通版本的小窗口吗？

如果可以，则在应用程序中使用现有控件，因为可以在以后再编写比现有控件更兼容的 API 定制小窗口。

- 小窗口一定要是可移植的吗？

如果开发者熟悉编写本机平台代码，则可以为应用程序支持的每个平台提供本机实现。如果它在任何（或者所有）平台上都不存在，则将还需要提供可移植实现。

- 可以通过扩展现有控件的行为来实现此功能吗？

如果可以的话，则考虑使用现有控件，并为它提供附加行为。

- 从用户的观点来看，开发者的小窗口将包含其他控件吗？

提前考虑此问题是很有用的，原因是它会影响小窗口的实现和 API。

一旦已经确定了需要定制小窗口并且已经决定了必须支持哪些平台，就可以考虑小窗口的几种实现技术，如本机实现、扩展现有小窗口或直接绘制小窗口。根据底层操作系统平台中提供的内容，这些技术可以配合使用。

1. 本机实现

如果应用程序需要不是由 SWT 提供的本机小窗口，则将需要在本地实现它。这可以是平台小窗口、第三方小窗口或者是平台共享库中的其他任何小窗口。

每个 SWT 平台都交付有共享库（例如 Windows 上的 DLL）和 JAR。共享库包含 SWT 需要的所有本机功能，但是，它并不意味着平台上可用的一组完整功能。要展示 SWT 未展示的本机功能或本机小窗口，需要编写开发者自己的共享库。如果开发者正在使用一个平台上的本机代码与另一个平台上的可移植代码的组合，则应确保在具有本机小窗口的平台上调用共享库，并在具有可移植小窗口的平台上调用 JAR。

要实现本机小窗口，则必须了解“Java 本机接口（JNI）”、共享库中的小窗口的 API 以及用 C 语言编写的底层操作系统平台 API。

实现的基本过程会决定在 Java API 中将展示本机小窗口的 API 的哪一部分，并编写 Java 代码来调用本机程序以实现该行为。必须编写 JNI C 语言代码才能调用共享库。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

2. 扩展现有小窗口

如果新的小窗口在概念或实现上类似于现有小窗口，则开发者可能想覆盖现有 SWT 小窗口。此技术用于实现 TableTree。

要覆盖小窗口，应创建组合体或者画布小窗口的子类。在定制小窗口的构造器中，创建覆盖的小窗口。生成的小窗口将是 Java 可移植的，原因是开发者正在调用覆盖的小窗口的 API 来进行实现。

通过覆盖小窗口来实现定制小窗口通常要比从头实现定制小窗口更简单。然而，需要注意以下几点。

- 考虑您的小窗口是否是一种被覆盖的小窗口，或者它只是使用一个被覆盖的小窗口来进行实现。如果开发者纯粹为了实现原因而覆盖小窗口，则开发者的 API 可能看起来不会类似于底层小窗口的 API。
- 尽可能少地转发方法和事件。不要重复实现小窗口的整个 API，否则，当覆盖的 API 在将来发行版中发生更改时，开发者将需要经常进行采用。应当转发大多数小窗口的公共方法，例如，setFont、setForeground 和 setBackground。
- 如果开发者发现自己正在实现大多数覆盖小窗口的 API，则应考虑在 API 级别展示覆盖的小窗口，并允许应用程序代码直接使用覆盖的小窗口。在此情况下作为“适配器”来实现功能部件可能会更好，它将行为添加到小窗口中，而不是将自己当做小窗口。
- 建议不要通过子类化小窗口来扩展小窗口，原因是它将使开发者的小窗口依赖于超类的实现。

3. 定制的绘制实现

在某些情况下，没有任何本机代码或现有小窗口来帮助开发者实现新的小窗口。这就意味着开发者自己必须使用 SWT 图形调用来绘制小窗口。尽管这可能会很复杂，但是这样实现的程序具有很好的可移植性。

使用下列规则来子类化画布或组合体类，可实现定制的已绘制的控件：

- 如果小窗口将没有任何子代，则子类化画布。这就意味着不允许应用程序创建小窗口的子代，并且不打算创建任何子代来实现小窗口。画布同时用于简单控件（例如，具有固定样式的标号）和较复杂的控件（例如，具有固定样式的文本编辑器）。在这两种情况下，小窗口完全是在内部使用图形调用来实现的，并且应用程序没有添加任何子代。
- 如果小窗口将具有来自应用程序的透视图的子代，或者开发者将创建子代来实现小窗口，则子类化组合体。当你正在组合小窗口以便创建新的小窗口时（例如，使用文本和列表来实现组合控件），就要使用组合体。当实现没有子代但允许客户机添加子代的小窗口时，也会使用它。ViewForm 控件也是这样。SashForm 小窗口代表两种情况：它在内部使用小窗口（窗框）来实现，并允许客户机添加它们自己的子代。

在已绘制的定制控件中，内部状态保存在 Java 实例变量中。根据小窗口的要求来定

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

义开发者的 API 和样式。已绘制的定制小窗口内部实现通常涉及下列主要任务。

- 创建构建器中需要的所有图形对象，并将它们存储在实例变量中。为画布或组合体上的销毁事件注册侦听器，以便在小窗口毁坏时可以释放这些对象。
- 将 paintListener 添加到画布或组合体中，并根据设计来绘制小窗口。对于复杂的小窗口，进行了许多工作来优化此过程（通过只计算和重新绘制绝对需要的内容）。
- 确保影响小窗口外观的任何 API 调用都将触发重新绘制小窗口。通常，当你知道必须进行重新绘制时，应该使用重新绘制来破坏小窗口，而不是直接调用内部绘制代码。这使平台有机会折叠你想利用任何其他挂起的绘制来生成的绘制，并通过使所有绘制都集中通过同一个位置来使代码更为流畅。
- 如果小窗口在它的 API 中定义事件，则确定哪些低级别画布或组合体事件将触发小窗口的事件。例如，如果程序中含有被单击事件，则需要在画布上注册鼠标事件并执行计算（例如，击中测试），以确定画布中的鼠标事件是否应该触发小窗口事件。

11.4 SWT 的布局

使用 SWT 技术，可以根据父代小窗口的大小来确定子代小窗口的大小或位置。SWT 定义了布局，它提供组合体中子代小窗口的通用定位和缩放。布局是抽象类布局的子类。可以在 org.eclipse.swt.layout 包中找到 SWT 标准布局。

布局是封装调整大小功能以便重复使用的有效方法。使用布局减少了应用程序代码。对于复杂的布局，简化效果尤其明显。SWT 提供了可用于许多情况的四个默认布局类，FillLayout、RowLayout、GridLayout 和 FormLayout。

11.4.1 FillLayout

填充布局 FillLayout 是最简单的布局类。填充布局在容器中以相同的大小、单行或单列地排列组件，强制它们全部使用相同的大小。最初，所有小窗口都将与最高的小窗口一样高，并且与最宽的小窗口一样宽。

可以使用填充布局来对任务栏或工具栏中的按钮进行布局，或者将复选框叠放在组中。当组合体只有一个子代时，还可以使用 FillLayout。填充布局的类型可以是 HORIZONTAL（行布局）或 VERTICAL（列布局）之一。

11.4.2 RowLayout

行布局 RowLayout 以单行或多行的方式使用几个选项，如 fill、wrap、spacing、justify、type 等定制组件的排列方式。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

行布局可以可选地覆盖小窗口，根据需要创建足够多的行或列来显示它们。行布局还提供布局的每个边缘上可配置的页边距，以及布局中的小窗口之间可配置的间距。开发者可以包装一个行布局，它强制所有小窗口使用相同大小。如果调整 RowLayout，则组合体中保留的额外空间将被分配给小窗口之间的页边距。表 11.3 总结了行布局的属性。

表 11.3 行布局的属性

属 性	描 述
调整	可以是行布局或列布局之一
底空白	指定要沿布局底边放置的像素数
左空白	指定要沿布局左边放置的像素数
右空白	指定要沿布局右边放置的像素数
顶空白	指定要沿布局顶部放置的像素数
包装	指定是否应强制所有小窗口具有相同大小
间隔	指定一个单元与其相邻单元之间的像素数
类型	可以是 HORIZONTAL（行布局）或 VERTICAL（列布局）之一
回绕	指定若在当前行或列中没有足够空间，是否将控件回绕到下一行或列

可以在 RowData 对象中指定 RowLayout 中每个小窗口的高度和宽度。表 11.4 总结了行数据的属性。

表 11.4 行数据的属性

属 性	描 述
宽度	指定以像素计的单元宽度
高度	指定以像素计的单元高度

11.4.3 GridLayout

网格布局 GridLayout 类似于 swing 的 GridLayout 的方式，以网格的方式排列组件。网格布局是比较强大和复杂的布局之一。网格布局在网格中布局小窗口，有许多可配置的参数，当调整组合体的大小时，用来控制网格行和列的缩放行为。

网格布局定义用来控制布局的总体战略的 API。最重要的属性为 numColumns，它确定网格的水平大小。通常，当首次设计窗口的外观时，就要决定此值。网格中小窗口的次序与开发者创建它们的次序相同。要更改网格中小窗口的次序，可以使用控件方法 moveAbove(Control)和 moveBelow(Control)。这些方法允许小窗口在布局中互相之间前后插入。表 11.5 总结了网格布局的可配置参数。

表 11.5 网格布局的可配置参数

属 性	描 述
horizontalSpacing	一个单元的右边缘与它的相邻单元的左边缘之间的像素数目
makeColumnsEqualWidth	指定是否应该强制所有列为相同宽度
marginWidth	用于网格的右边缘和左边缘上页边距的像素数目

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

续表

属 性	描 述
marginHeight	用于网格顶部和底部边缘上页边距的像素数目
numColumns	应该用来生成网格的列数
verticalSpacing	一个单元的底部边缘与它相邻单元的顶部边缘之间的像素数目

网格布局支持网格中每个小窗口的许多其他布局参数。这些属性是在网格数据（GridData）对象中指定的。必须将网格数据设置为网格中每个小窗口的布局数据。

网格数据类定义样式常量，开发者可在网格数据构造器中指定布局参数的常用组合。还可以使用网格数据中的公共方法来单独设置这些属性。

通过使用网格数据可以获得一些高动态和复杂的布局。将不同小窗口的这些值组合在网格中可以生成布局的许多不同组合。因为允许小窗口跨越单元，所以许多布局并不是严格意义上的网格。表 11.6 总结了网格数据的可配置参数。

表 11.6 网格数据的可配置参数

属 性	描 述
grabExcessHorizontalSpace	指定单元是否应该增大，以便使用网格中可用的任何额外水平空间。在根据小窗口和它们的网格数据计算网格中的单元大小之后，将把“组合体”中余下的任何额外空间分配给占用额外空间的那些单元
grabExcessVerticalSpace	指定单元是否应该增大，以便使用网格中可用的任何额外垂直空间
heightHint	指定小窗口的最低高度（以及包含小窗口的行的最低高度）
horizontalAlignment	可以是 BEGINNING、CENTER、END 和 FILL 的其中之一。FILL 意味着将调整小窗口的大小，以便占据它的网格单元的整个宽度
horizontalIndent	小窗口与它的网格单元的左边缘之间的像素数目
horizontalSpan	指定小窗口应该跨越的网格中的列数。默认情况下，小窗口占据网格中的一个单元。它可以通过增大此值来水平获取其他单元
verticalAlignment	可以是 BEGINNING、CENTER、END 和 FILL 的其中之一。FILL 意味着将调整小窗口的大小，以便占据它的网格单元的整个高度
verticalSpan	指定小窗口应该跨越的网格中的行数。默认情况下，小窗口占据网格中的一个单元。它可以通过增大此值来垂直获取其他单元
widthHint	指定小窗口的最小宽度（以及包含小窗口的列的最小宽度）

11.4.4 FormLayout

表单布局 FormLayout 是 SWT 2.0 的新特性，通过定义组件四个边的“粘贴”位置来排列组件，被引用的相对组件可以是父组件，也可以是同一容器中的其他组件。表 11.7 总结了表单布局的属性。

表 11.7 表单布局的属性

属 性	描 述
marginHeight	指定将对布局顶和底边设置的垂直页边距的像素数目
marginWidth	指定将对布局左和右边设置的水平页边距的像素数目

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

表单布局中的左右页边距用 marginWidth 定义，上下用 marginHeight 定义。边距也可以根据附件中的每个控件标准来定义，默认情况下上面两域为 0。

例如，下面的代码在父 Composite（这里是 Shell）的四周都设置边距为 5 个像素。

```
Display display = new Display ();
Shell shell = new Shell (display);
FormLayout layout= new FormLayout ();
layout.marginHeight = 5;
layout.marginWidth = 5;
shell.setLayout (layout);
```

表单布局使用附件模型来布局小窗口。可以为小窗口的每侧定义一个表单附件（FormAttachment）并将其存储在它的表单数据（FormData）中。表单数据可以用来指定小窗口请求的宽度和高度，只要这些值不与附件约束发生冲突。表 11.8 总结了表单数据的属性。

表 11.8　表单数据的属性

属　性	描　述
顶部	指定控件顶部的附件
左边	指定控件左边的附件
底部	指定控件底部的附件
右边	指定控件右边的附件
宽度	指定表单中控件的首选宽度（以像素计）
高度	指定表单中控件的首选高度（以像素计）

表单数据对象指定了表单布局中的每个控件如何布局。每个控件的表单对象定义了它四个边的附件值，根据这些附件值来放置控件的每个边。可以使用控件的 setLayoutData 方法为控件设置表单数据对象，例如：

```
Button button1 = new Button(shell, SWT.PUSH);
button1.setText("B1");
button1.setLayoutData(new FormData());
```

上面的这段代码并没有为控件的 FormData 对象设置附加值。在这种情况下，则使用默认的附加值。

表单附件用于将指定的小窗口一侧连接到父代组合体或布局中的另一个小窗口。通常不必设置小窗口所有侧面上的附件。普遍的做法是仅指定一个水平（左或右）附件和一个垂直（顶部或底部）附件，并允许小窗口采用在它们的表单数据中指定的大小或其首选大小，如果在表单数据中未指定大小的话。表 11.9 总结了表单附件的属性。

表 11.9　表单附件的属性

属　性	描　述
对齐	指定与控件连接的控件端对齐。DEFAULT 指示应将小窗口连接到指定控件的相邻一侧。对于顶部和底部附件，使用 TOP、BOTTOM 和 CENTER 来指示小窗口的指定一侧连接到控件的指定一侧。对于左边和右边的附件，使用 LEFT、RIGHT 和 CENTER 指示小窗口的指定一侧连接到控件的指定一侧。例如，使用 TOP 指示附件小窗口的顶部应连接到指定的控件的顶部。）

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

续表

属 性	描 述
控件	指定附件小窗口连接至的目标控件
分母	指定定义附件的等式 y=ax+b 中"a"项的分母
分子	指定定义附件的等式 y=ax+b 中 "a"项的分子
偏移量	指定控件侧从附件位置的偏移量(以像素计)，可以是正或负。这是定义附件的等式 y=ax+b 中的"b"项

左边、右边、顶部和底部这四个域给出了控件的左右上下这四个边，下面是例子代码：

```
FormData formData = new FormData();
formData.top = new FormAttachment(0,50);
formData.bottom = new FormAttachment(100,-5);
formData.left = new FormAttachment(20,0);
formData.right = new FormAttachment(100,-3);
button1.setLayoutData(formData);
```

表单附件对象定义了控件指定边的附加值。可以用以下各种方法配置附件。

- 将小窗口的已指定一侧连接到父代小窗口中的位置（百分比值 0~100）。可以添加可选的像素偏移量。
- 将小窗口的已指定一侧连接到组合体中目标小窗口的相邻一侧。可以添加可选像素偏移量。根据源小窗口中指定的一侧推算出连接的目标小窗口的一侧。例如，连接顶边到小窗口意味着它被连接到相邻小窗口的底边。
- 将小窗口指定的一侧连接到组合体中目标小窗口的特定一侧。可以添加可选的像素偏移量。例如，这可以用来将小窗口的顶部与另一个小窗口的顶部对齐，这样它们将总是垂直对齐。
- 使小窗口在目标小窗口中居中。只需为正在确定中心方向的一侧（左边或右边、顶部或底部）指定中心。

在 SWT 中，可以由用户自定义布局类。可以利用现有布局或者通过嵌套几个不同的布局来实现布局。如果开发者认为需要定制布局类，则最好是首先在调整大小侦听器中实现布局算法。这使得算法本身的调试更简单。一定要测试布局的各种情况，如调小、调大、覆盖和剪辑等。一旦算法生效，就可以将代码重构成布局的子类。

11.5 SWT 的事件和线程

11.5.1 事件

每当从队列中读取事件并将它调度给小窗口时就会进行实际的工作。大多数应用程序逻辑是作为对用户事件的响应实现的。

基本模式是将侦听器添加至已经创建的一些小窗口，并且当发生适当的事件时，执

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

行侦听器代码。

对于每种类型的侦听器，都有一个接口来定义侦听器，由类来提供事件信息，由 API 方法来添加侦听器。如果在侦听器接口中定义了多种方法，则会提供适配器来实现侦听器接口并同时提供空方法。所有事件、侦听器和适配器都是在包 org.eclipse.swt.events 中定义的。

类型化事件系统是利用低级的非类型化小窗口事件机制来实现的。此机制并不是用来供应用程序使用的，但是将看到它在 SWT 实现的内部使用。它还用于许多工作台向导页面实现。

非类型化机制依赖于常量来标识事件类型，并定义随此常量一起提供的一般侦听器。它允许侦听器实现“事例样式”侦听器。在以下代码片段中，定义了一般事件处理程序，并向 shell 中添加了几个侦听器。

```
Shell shell = new Shell();
Listener listener = new Listener() {
   public void handleEvent(Event e) {
      switch (e.type) {
         case SWT.Resize:
            System.out.println ("Resize received");
              break;
         case SWT.Paint:
            System.out.println ("Paint received");
            break;
         default:
            System.out.println("Unknown event received");
      }
   }
};
shell.addListener(SWT.Resize, listener);
shell.addListener(SWT.Paint, listener);
```

11.5.2 线程

当使用小窗口工具箱时，了解操作系统的图形用户界面事件的底层线程模型是很重要的。

在任何图形用户界面应用程序中，不管它的编程语言或用户界面工具箱是什么，操作系统平台都会检测图形用户界面事件，并将它们放在应用程序事件队列中。尽管在不同的操作系统平台上机制稍微有些不同，但是基础是相似的。当用户单击鼠标、输入字符或者对窗口的外观进行处理时，操作系统将生成应用程序图形用户界面事件，例如，鼠标单击、击键或者窗口绘制事件。它确定哪个窗口和应用程序应当接收每个事件，并将它放置在应用程序的事件队列中。

任何窗口化的图形用户界面应用程序的底层结构都是事件循环。应用程序进行初始化，然后启动循环，它只从队列中阅读图形用户界面事件，并相应地作出反应。在处理其中一个事件时完成的任何工作都必须快速地进行，以便让图形用户界面系统可以响应用户。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

在任何操作系统平台的图形用户界面系统中，对组件或一些图形 API 的访问操作都要被严格同步并串行化。例如，在一个图形界面中的按键组件可被设成可用状态或禁用状态，正常的处理方式是，用户对按键状态的设置操作都要被放入到图形用户界面系统的事件处理队列中，这意味着访问操作被串行化，然后依次处理，这意味着访问操作被同步。想像当按键可用状态的设置函数还没有执行结束的时候，程序就希望再设置该按键为禁用状态，势必会引起冲突。实际上，这种操作在任何图形用户界面系统都会触发异常。

应该在单独的线程中执行由用户界面事件触发的长时间操作，以便允许事件循环线程快速返回，并从应用程序的队列中访存下一个事件。然而，必须利用显式锁定和序列化来控制从其他线程访问小窗口和平台 API。未能遵循规则的应用程序可能会导致操作系统调用失败，更糟糕的是，可能会锁定整个图形用户界面系统。

Java 语言本身就提供了多线程机制，这种机制对图形用户界面编程来说是不利的，它不能保证图形组件操作的同步与串行化。

用 Java 编写的较高级别的小窗口工具箱通常试图通过隐藏平台事件循环来向应用程序开发者隐藏用户界面线程问题。实现此过程的常见方法是设置专用工具箱用户界面线程，以便读取和调度事件循环，并将事件送至正在单独线程中运行的应用程序服务的内部队列中。这允许工具箱有足够的时间来响应操作系统，而不对应用程序处理事件的时间设置任何限制。应用程序必须仍然使用特殊的锁定技术来从它们的应用程序线程中访问用户界面代码，但是，它在整个代码中的完成是一致的，原因是所有应用程序代码正在非用户界面线程中运行。

但是，这样它将导致许多问题。当图形用户界面事件的时间取决于 Java 线程的实现和应用程序性能时，就很难调试和诊断问题。

目前的图形用户界面平台对事件队列执行了许多优化。常见的优化就是将连续的绘制事件折叠到队列中。每当必须重新绘制窗口的一部分时，就可以检查队列是否存在绘制事件重叠或尚未调度的冗余绘制事件。可以将这些事件合并到一个绘制事件中，从而使得较少出现闪烁，并且也不会太频繁执行应用程序的绘制代码。如果小窗口工具箱正在将事件快速拉出队列，并将它们发送到内部队列中，则此优化将会失败。

SWT 遵循平台直接支持的线程模型。应用程序在它的主线程中运行事件循环，并直接从此线程中调度事件。这就是应用程序的用户界面线程。只有在这个惟一线程中才能调用对组件或某些图形 API 的访问操作。如果在非用户线程中程序直接调用这些访问操作，那么 SWTExcepiton 异常会被抛出。从而满足了本地图形用户界面系统对线程的要求。

从技术观点看，用户界面线程是创建显示的线程。实际上，此线程也是运行事件循环和创建小窗口的线程。

由于所有事件代码都是从应用程序的用户界面线程中触发的，因此，处理事件的应用程序代码可以自由地访问小窗口，并且不需要任何特殊技术就可以进行图形调用。然而，当响应某事件而执行长时间运行的操作时，应用程序负责创建计算线程的分支。SWT 应用程序的主线程（包括事件循环）如下所示：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
public static void main (String [] args) {
   Display display = new Display ();
   Shell shell = new Shell (display);
   shell.open (); // 启动事件循环
   //清除窗口.
   while (!shell.isDisposed ()) {
      if (!display.readAndDispatch())
         display.sleep();
   }
   display.dispose ();
}
```

一旦创建了小窗口并且打开了 Shell，应用程序就会读取和调度操作系统队列中的事件，直到清除 shell 窗口为止。如果队列中没有事件可用，则说明显示进入睡眠状态，以使其他应用程序有机会运行。

SWT 应用程序最常见的线程模型是在计算线程中运行单个用户界面线程并执行长时间操作。然而，SWT 不会限制开发者使用此模型。应用程序可以运行多个用户界面线程，而每个线程中都有单独的事件循环。SWT 提供了特殊的访问方法来从后台线程中调用小窗口和图形代码。

SWT 也在 org.eclipse.swt.widget.Display 类中提供了可以间接地在非用户线程中进行图形组件的访问操作，这是通过 syncExec(Runnable)和 asyncExec(Runnable)这两个方法实现的。

- 当非用户界面线程中的应用程序代码取决于用户界面代码的返回值时，应当使用 syncExec(Runnable)，否则，需要确保可运行程序在返回到线程中之前能够完成运行。SWT 将在从应用程序的用户界面线程运行完可运行程序之前阻塞调用线程。例如，正在根据窗口的当前大小来计算某些内容的后台线程将要同步地运行代码以获取窗口的大小，然后继续它的计算。
- 当应用程序需要执行某些用户界面操作，但是与继续之前需要完成的操作无关时，应当使用 asyncExec(Runnable)。例如，更新进度指示器或者重绘窗口的后台线程可以请求异步更新，并继续它的处理。在此情况下，后台线程的时间与可运行程序的执行之间没有承诺关系。

方法 syncExec()和 asyncExec()的区别在于前者要在指定的线程执行结束后才返回，而后者则无论指定的线程是否执行都会立即返回到当前线程。以下代码片段演示了使用这些方法的模式：

```
// 应用程序的执行与所要完成的操作无关，因此使用异步方式
Display.getDefault ().asyncExec (new Runnable () {
   public void run() {
      myWindow.redraw();
   }
});
// 下面是其他处理
...
```

当开发者从头实现 SWT 应用程序时，由开发者控制事件循环的创建及决定应用程序中计算线程的分支，规则如下。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- 工作台插件代码在工作台的用户界面线程中执行。
- 如果从工作台中接收到事件，则它始终是在工作台的用户界面线程中执行。
- 如果插件将计算线程分支，则在对工作台、JFace 或 SWT 调用任何 API 时，它必须使用显示 asyncExec(Runnable)或 syncExec(Runnable)方法。
- 工作台和 JFace API 调用不会检查调用者是否正在用户界面线程中执行。然而，对于从非用户界面线程执行的所有 API 调用，SWT 将触发 SWTException。
- 如果插件使用 JFace IRunnableContext 接口来调用进度指示器和运行操作，则它会提供一个自变量来指定是否对计算线程分支以便运行操作。

11.6　开发 SWT 程序的示例

在 Eclipse 中开发 SWT 应用程序和开发其他应用程序一样，首先需要创建一个 Java 项目和包，假设项目名和包名分别为 SWTProject 和 SWTPackage，并在新建 Java 项目的 Java 设置对话框的“库”选项卡中将 SWT 库添加项目的构建路径中，添加步骤如下。

① 选中“库”选项卡。（如已创建项目，也可在项目的上下文菜单中选择“属性”，然后在“属性”对话框左侧选中“Java 构建路径”类别，然后在右侧窗格中选中“库”选项卡。）

② 单击“添加变量”按钮，这将打开“新的变量类路径条目”对话框窗口，如图 11.1 所示。

③ 在对话框窗口选中“ECLIPSE_HOME”条目，然后单击“扩展”按钮，这将打开“变量扩展”窗口，如图 11.2 所示。

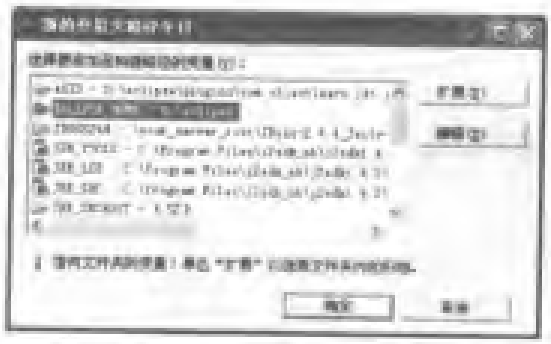


图 11.1　“新的变量类路径条目”对话框窗口



图 11.2　“变量扩展”窗口

④ 在“变量扩展”窗口逐层展开“plugins”子目录，定位到 swt.jar 所在的目录，通常为 plugins 子目录下的\org.eclipse.swt.win32_2.1.0\ws\win32 目录（但也可能随着 Eclipse 的版本不同而有所不同）。

⑤ 依次单击“确定”按钮，完成项目的类路径条目的添加。

现在，可以创建 SWT 应用程序了。在包中新建一个 Java 类，假设类名为 SWTTest.java，其源代码假设如下：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
package SWTPackage;
import org.eclipse.swt.layout.RowLayout;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Shell;

public class SWTTest {
    public static void main(String args[]) {
        // 创建一个显示
Display display = new Display();

        // 创建一个 Shell
Shell shell = new Shell(display);

        // 设置布局
shell.setLayout(new RowLayout());

        // 压缩窗口
shell.pack();

        // Shell 窗口必须打开才能显示
shell.open();

        // 事件处理循环读取并转发 GUI 事件
while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }

        // 在 Shell 被丢弃的时候销毁 Display
        display.dispose();
    }
}
```

这个例子显示了一个空的窗口，其程序各部分的解释见代码注释。现在可以试试运行程序了，然而遗憾的是并没有所期待的空窗口出现，而是在控制台中显示运行时异常，如图 11.3 所示。

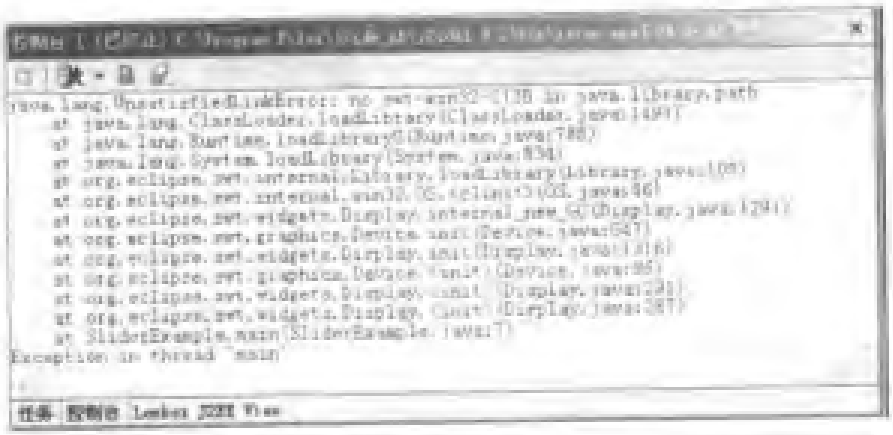


图 11.3　运行时异常

因为 SWT 应用了 JNI 技术，因此同时也要找到相对应的 JNI 本地化库文件，从而运行 SWTTest.java 程序会抛出“no swt-win32-2133 in java.library.path”的异常信息。为了解决这个问题，需要设置执行程序的 Java 虚拟机的运行参数 java.library.path，从而在程

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

序运行时使用本地库。在本例中，本地库为 Eclipse 安装目录（d:\eclipse）下的文件：

```
\plugins\org.eclipse.swt.win32_2.1.2\os\win32\x86\swt-win32-2135.dll
```

由于版本和操作系统平台的不同，本地化库文件的名称会有些差别。在 Eclipse 中，设置程序运行时的 Java 虚拟机变量的步骤如下。

① 选中项目，然后在 Eclipse 工作台主菜单中选择“运行”→“运行”，这将打开“运行”窗口。

② 在“运行”窗口中选中“(x)=自变量”选项卡。

③ 如图 11.4 所示，在该选项卡下的“VM 自变量”窗格中输入如下内容：

```
java -Djava.library.path=d:\eclipse\plugins\org.eclipse.swt.win32_2.1.0\
os\win32\x86
```



图 11.4　设置本地库目录

④ 依次单击“应用”和“运行”按钮。

通常情况下，此时程序将能正常运行。但在有些 Eclipse 版本和操作系统平台上，执行程序时仍然无法找到本地库文件。这时一种妥善的解决方法是将本地库文件（如本例中的 d:\eclipse\plugins\org.eclipse.swt.win32_2.1.0\os\win32\x86\swt-win32-2135.dll）拷贝到其他任一目录中，并将该目录如上设置为本地库目录。

程序的运行结果如图 11.5 所示，为一空白窗口，并将其初始状态设置为压缩状态。



图 11.5　程序的运行结果显示

下面在该程序的基础上构建一个稍复杂一些的 SWT 程序，程序涉及了如何应用

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Color 等系统资源，以及如何释放系统资源等。程序的具体解释参见程序中的详细注释语句。

```
package SWTPackage;
import org.eclipse.swt.*;
import org.eclipse.swt.layout.*;
import org.eclipse.swt.widgets.*;
import org.eclipse.swt.graphics.*;
import org.eclipse.swt.events.*;

public class SWTTest {
    public static void main(String args[]) {

        //创建一个显示
        Display display = new Display();

        //创建一个 Shell
        Shell shell = new Shell(display);

        //调用窗口初始化方法
        initShell(shell);

        //压缩窗口
        shell.pack();

        //Shell 窗口必须打开才能显示
        shell.open();

      //事件处理循环读取并转发 GUI 事件
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }

      //在 Shell 被丢弃的时候销毁 Display
        display.dispose();
    }

    private static void initShell(Shell shell) {

        //为 Shell 设置布局对象
        GridLayout gShellLay = new GridLayout();
        shell.setLayout(gShellLay);

        //构造一个 Composite 控件作为文本框和按键的容器
        Composite panel = new Composite(shell, SWT.NONE);

        //为 Panel 指定一个布局结构对象。
        //这里让 Panel 尽可能的占满 Shell，也就是全部应用程序窗口的空间。
        GridData gPanelData =
            new GridData(
                GridData.GRAB_HORIZONTAL
                    | GridData.GRAB_VERTICAL
                    | GridData.FILL_BOTH);
        panel.setLayoutData(gPanelData);

        //为 Panel 也设置一个布局对象。文本框和按键将按这个布局对象来显示。
        GridLayout gPanelLay = new GridLayout();
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
        panel.setLayout(gPanelLay);

        //为 Panel 生成一个背景色
        final Color bkColor = new Color(Display.getCurrent(), 200, 0, 200);
        panel.setBackground(bkColor);

        //生成文本框
        final Text text = new Text(panel, SWT.MULTI | SWT.WRAP);

        //为文本框指定一个布局结构对象，这里让文本框尽可能的占满 Panel 的空间。
        GridData gTextData =
            new GridData(
                GridData.GRAB_HORIZONTAL
                    | GridData.GRAB_VERTICAL
                    | GridData.FILL_BOTH);
        text.setLayoutData(gTextData);

        //生成按键
        Button butt = new Button(panel, SWT.PUSH);
        butt.setText("Push");

        //为按键指定鼠标事件
        butt.addMouseListener(new MouseAdapter() {

            public void mouseDown(MouseEvent e) {

                //当用户单击按键的时候，显示信息
                text.setText("Hello, SWT");
            }
        });

        //当主窗口关闭时，会触发 DisposeListener，这里用来释放 Panel 的背景色。
        shell.addDisposeListener(new DisposeListener() {

            public void widgetDisposed(DisposeEvent e) {
                bkColor.dispose();
            }
        });
    }

}
```

该程序含有一个文本框和一个按钮，如图 11.6 所示。当用户单击按钮的时候，文本框显示一句欢迎信息“Hello，SWT”，如图 11.7 所示。

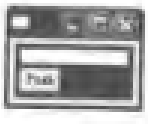


图 11.6　程序运行结果



图 11.7　单击“Push”后的显示

11.7　JFace 简介

JFace 是插件的用户界面框架。JFace 是一个用户界面工具箱，它提供很难实现的、

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

用于开发用户界面功能部件的 helper 类。JFace 在原始的小窗口系统的级别之上运行。通过 JFace，开发者可以专注于实现特定插件的功能，而不必花费精力来处理底层小窗口系统或者解决几乎在任何用户界面应用程序中都很常见的问题。

JFace 提供用于处理常见的用户界面编程任务的类：

- 查看器负责处理填充、排序、过滤和更新小窗口等。
- 操作和添加项介绍用于定义用户操作的语义，并指定在何处提供它们。
- 图像和字体注册表提供用于处理用户界面资源的常见模式。
- 对话框和向导定义用于与用户进行复杂交互的框架。

JFace 的 API 比 SWT 更加易于使用，但功能却没 SWT 来的直接。比如下面的代码应用 JFace 中的 MessageDialog 打开一个警告对话框：

```
MessageDialog.openWarning(parent, "Warning", "Warning message");
```

如果使用实现 SWT 实现同样的功能，如前面小节中介绍的，程序将要复杂得多。因此，对开发人员来说，在开发一个图形组件的时候，比较好的方式是先到 JFace 包去找一找，看是不是有更简洁的实现方法，如果没有再用 SWT 包去实现。

11.8 JFace 的体系结构

JFace 应用程序窗口是顶级窗口的 JFace 类。所谓顶级窗口是指由操作系统窗口管理器管理的窗口。JFace 窗口实际上不是顶级窗口的图形用户界面对象。相反，JFace 窗口是助手对象，它知道对应的 SWT Shell 对象，并提供代码来帮助创建 / 编辑它，以及侦听它的事件等。

JFace 并不是 SWT 之上的层，它并不向开发者隐藏 SWT。相反，JFace 有几种使用 SWT 的常用模式，而且它提供一些使用程序代码，帮助用户更方便地对这些模式进行编程。应用程序、JFace 和 SWT 之间的关系如图 11.8 所示。

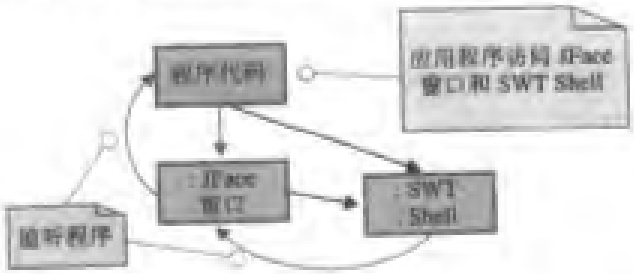


图 11.8　应用程序、JFace 和 SWT 间的关系

SWT 和 JFace 之间的界限是很明显的。SWT 根本不依赖于任何 JFace 或平台代码。JFace 用来在 SWT 库顶部提供常见的应用程序用户界面功能。JFace 并不试图“隐藏”SWT 或者替换它的功能。它提供一些类和接口，以处理与使用 SWT 来对动态用户界面编程相关联的许多常见任务。

JFace 也开和 SWT 一样有着完整独立的开发包。JFace 开发包的包前缀是 org.eclipse.

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com

jface。JAR 包和源代码也和 SWT 一样，保存在 Eclipse 安装路径下的\plugins 子目录下。

通过定义工作台的扩展点即可使用插件添加平台的用户界面功能。许多这些扩展点，特别是向导扩展，是通过使用 org.eclipse.jface.org.eclipse.swt 包中的类来实现的。工作台使用 JFace，但是又试图尽可能减少依赖项。例如，工作台部件模型 IWorkbenchPart 就设计为独立于 JFace。例如，可以直接使用 SWT 小窗口来实现视图和编辑器，而不必使用任何 JFace 类。

工作台尽可能保持“JFace 中立”，允许开发人员使用他们觉得有用的 JFace 的部件。实际上，在工作台的大多数实现中都使用了 JFace，并且在 API 定义中引用了 JFace 类型。何处是 JFace 结束而工作台开始的位置？有时候界线并不是很明显。通常，来自于包 org.eclipse.jface.org.eclipse.swt 的 JFace API 与工作台扩展和 API 无关。

11.9　查看器

JFace 为 SWT 中的大多数重要的小窗口提供查看器。使用查看器可以实现工作台视图和编辑器。查看器通常用于列表、树、表和文本小窗口。

面向列表的查看器保存域对象（称为元素）的列表，并将这些列表显示在其相应的 SWT 小窗口中。列表查看器知道如何从列表中的任何元素中获取文本标号。文本小窗口具有许多常见语义，例如，双击行为、撤销、着色及按索引或者行来导航。TextViewer 是 SWT StyledText 小窗口的适配器。文本查看器为客户提供了文档模型，并管理从文档至由文本小窗口提供的样式文本信息的转换。

从前面所述可知，可以利用 SWT 小窗口来直接实现工作台用户界面添加项，例如，视图、编辑器等，那为什么还要使用查看器？

查看器允许开发者在创建小窗口时仍然可以使用模型对象。如果直接使用 SWT 小窗口，则需要将对象转换成 SWT 期望的字符串和图像。查看器充当 SWT 小窗口上的适配器，处理用于处理小窗口事件的常见代码，否则，将需要由开发者自己来实现。

查看器是某些 SWT 部件的基于模型的适配器。查看器处理公用行为并提供比 SWT 部件所提供的更高层的语义。列表、树和表的标准查看器支持发布带客户域元素的查看器并保持与该域修改的同步。这些查看器配有内容供应者和标签供应者。内容供应者知道如何将查看器的输入元素映射到期望的查看器内容，以及如何成功地将域的修改对应到相关的查看器更新上。标签供应者知道如何提供部件中显示任一给定域元素所需的字符串标签和图标。

每个查看器都有相关联的 SWT 小窗口。可以通过查看器构造函数中提供父组合体来隐式地创建此小窗口，或者通过首先创建它，然后将它提供给它的构造函数中的查看器，从而显式地创建它。

查看器可以有选择地配置基于元素的过滤器和分类器。根据提供给查看器的域元素，客户得到相应选择和事件的通知。查看器实现处理域元素和 SWT 部件之间的映射，调整元素已过滤的视图并在必要时重新分类。标准文本查看器支持常见的操作如双击操

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

作、取消、着色及按字符索引或行号浏览。文本查看器为客户提供一个文档模型，并管理文档到 SWT 文本部件所需信息之间的转换。多个查看器可在同一模型或文档上打开；当其中任一模型或文档发生改变则自动更新所有的查看器。

要了解查看器，开发者必须熟悉查看器的输入元素、内容、选项以及在它正在处理的小窗口中实际显示的信息之间的关系。

输入元素是查看器正在显示（或编辑）的主要对象。从查看器的角度来看，输入元素可以是任何对象。不建议由输入元素来实现特定接口。查看器必须能够处理输入元素的更改。如果在查看器中设置了新的输入元素，则它必须根据新元素来重新填充它的小窗口，并取消它自己与前一输入元素的关联。对于每种类型的查看器，注册为输入元素上的侦听器的语义和根据元素填充小窗口的语义是不同的。

内容查看器是这样一种查看器：具有严格定义的协议，用来获取来自于它的输入元素的信息。内容查看器使用两个专门的 helper 类 IContentProvider 和 ILabelProvider 来填充它们的小窗口和显示关于输入元素的信息。

IContentProvider 提供了基本的有效期协议，以将内容提供程序与输入元素相关联以及处理输入元素的更改。为不同类型的查看器实现了多个专门的内容提供程序。最常用的内容提供程序是 IStructuredContentProvider，它可以提供给定输入元素的对象列表。它用于类似列表的查看器，例如，列表、表或树。通常，内容提供程序知道如何在输入元素和期望的查看器内容之间进行映射。

给定查看器的内容（派生自输入元素和内容提供程序），ILabelProvider 可以生成在查看器中显示内容所需的特定用户界面元素，例如，名称和图标。标号提供程序可以帮助保存图标资源，因为它们可以确保图标的相同实例用于查看器中所有相似类型。

输入元素、内容提供程序和标号提供程序允许查看器隐藏有关填充小窗口的实现的大部分详细信息。查看器的客户只需要关心是否使用了正确类型的输入和内容提供程序来填充查看器。标号提供程序必须知道如何从查看器内容中派生出用户界面信息。

11.10 JFace 操作和添加项

操作表示可以由最终用户触发的命令。操作通常与按钮、菜单项和工具栏中的项相关联。尽管操作本身不放置在用户界面中，但是它们具有面向用户界面的属性，例如，工具提示文本、标号文本和图像。这允许其他类为操作的表示构造小窗口。

每当在插件中定义命令时，就将实现操作类。如果向其他视图和编辑器添加操作，则将实现操作代理。工作台操作代理是利用与它们相关联的视图或编辑器来初始化的。

当用户触发用户界面中的操作时，就会调用该操作的运行方法来执行实际工作。运行方法中的一种常见模式是查询工作台选择并处理所选的对象。另一种常见模式是在选择操作时启动向导或对话框。

开发者不必直接实现 IAction 接口。而是应该子类化操作类。操作类允许开发者定义用户命令而不用考虑它们在用户界面中的表示。这使开发者能够灵活地更改操作在插

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

件中的表示，不必在选择某个命令时更改实际执行该命令的代码。添加项类用来管理表示命令的实际用户界面项。开发者不需要对添加项类编程。

操作机制使得用户命令可独立于它们在用户界面中的定义。JFace 查看器和操作实现了应用程序与小窗口之间的普通交互作用。一个操作代表一个命令，可由用户通过按钮、菜单项或工具条上的一项来触发。每个操作知道自己的关键用户属性（标签、图标、工具提示等），这些属性用于构建合适的部件来表现该操作。这种分离使得同一操作可用于用户界面的不同位置，并且也意味着可方便地改变操作在用户界面的位置而不用修改操作本身的代码。

添加项（IContributionItem）表示操作的用户界面部分。更准确地说，它表示添加到共享用户界面资源的项，例如，菜单或工具栏。添加项知道如何利用表示该添加的相应 SWT 项来填充特定的 SWT 小窗口。

当为工作台用户界面添加操作时，不必担心创建添加项。当工作台为已定义的操作创建用户界面项时，已自动完成了相关处理。

添加项管理器（IContributionManager）表示将在用户界面中提供添加项的集合。可以通过使用已命名的添加标识来添加和插入添加项，以便按照相应的次序来放置这些添加项。还可以根据标识来查找添加项和除去个别项。

IContributionManager 的每种实现都知道如何使用 SWT 小窗口的项来填充该小窗口。JFace 为菜单（IMenuManager）、工具栏（IToolBarManager）和状态行（IStatusLineManager）提供了添加项管理器。

插件开发者不需要实现这些接口，但是，在 API 方法中，将看到对其中某些管理器的引用。

11.11 在 Eclipse 中创建 JFace 应用程序

下面我们构建一个常见的“Hello World”程序。通过该程序，讨论在 Eclipse 中创建 JFace 应用程序。

Hello 类的代码如下：

```
import org.eclipse.jface.window.*;
import org.eclipse.swt.*;
import org.eclipse.swt.widgets.*;

public class HelloWorldClass extends ApplicationWindow {

    public HelloWorldClass() {
        super(null);
    }

    protected Control createContents(Composite parent) {
        Button b = new Button(parent, SWT.PUSH);
        b.setText("Hello World!");
        return b;
    }
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    public static void main(String[] args) {
        HelloWorldClass w = new HelloWorldClass();
        w.setBlockOnOpen(true);
        w.open();
        Display.getCurrent().dispose();
    }
}
```

下面，首先对前面的程序进行解释。

```
public class HelloWorldClass extends ApplicationWindow
```

该语句表明 HelloWorldClass 是 ApplicationWindow 的子类。ApplicationWindow 是顶层应用程序窗口，可以对其进行子类化，也可以加入特定行为。

```
HelloWorldClass w = new HelloWorldClass();
```

该语句将创建 ApplicationWindow 的一个子类。

```
w.setBlockOnOpen(true);
w.open();
setBlockOnOpen(true)使 open()堵塞，直到窗口关闭为止。

protected Control createContents(Composite parent){
    Button b = new Button(parent, SWT.PUSH);
    b.setText("Hello World!");
    return b;
}
```

该方法创建一个带有文本“Hello, World!”的按钮，该按钮将显示在内容（Contents）区域。参数 parent 表示内容区域的复合窗口组件。ApplicationWindow 将在所有其他窗口组件已经创建，但在屏幕显示窗口之前调用该方法。

需要注意的是，为了在 Eclipse 中正常运行该程序，首先需要将一些 JFace 程序必须的 JAR 文件置于类路径中，其步骤如下。

① 右键选中 hello 类所在的项目。

② 从弹出式上下文菜单中选择“属性”，将打开该项目的属性窗口。

③ 在属性窗口中选择 Java 构建路径，然后在右边窗格中选择“库”选项卡，如图 11.9 所示。

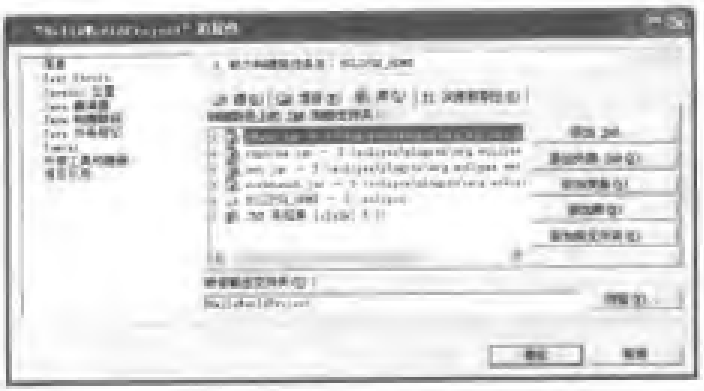


图 11.9 项目的属性窗口

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

④ 在“库”选项卡下，单击“添加外部 JAR(X)...”按钮，将以下 Eclipse 安装目录下的以下 JAR 文件添加到项目的构建路径上：

```
\plugins\org.eclipse.core.runtime_2.1.0\runtime.jar
\plugins\org.eclipse.ui.workbench_2.1.0\workbench.jar
\plugins\org.eclipse.jface_2.1.0\jface.jar
\plugins\org.eclipse.swt.win32_2.1.0\ws\win32\swt.jar
```

⑤ 单击“确定”，完成类路径设置。

此外，为了确保 Java 虚拟机正确获取运行时使用的图形用户界面共享库，需要执行以下步骤。

① 选中 hello 类所在的项目。

② 选择菜单“运行”→“运行（N）…”，将显示“运行（N）”窗口。

③ 在“运行（N）”窗口中，确认在左边窗格的对象树中高亮显示的为 hello 类所在的项目。

④ 在右边窗格中，选中“(x=)自变量”选项卡。

⑤ 在该选项卡下的“VM 自变量（G）：”栏目中输入：-Djava.library.path =D:\eclipse\plugins\org.eclipse.swt.win32_2.1.2\os\win32\x86\，如图 11.10 所示。



图 11.10　配置 Java 虚拟机的运行参数

⑥ 单击“关闭”按钮，完成 Java 虚拟机运行参数的设置。

此时运行该程序，则显示结果如图 11.11 所示。



图 11.11　包含单一按钮的窗口

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 12 章　Eclipse 下的 Struts 应用开发

Struts 提供了一个构建基于模型—视图—控制器（Model-View-Controller）体系结构的 Web 应用程序的框架。Struts 框架是一种基于 Java 的技术，Web 应用程序开发人员通过 Struts 框架即可利用面向对象设计、代码重用技术构建可扩展、可重用的软件。Struts 的目标是使得用可重用组件构建和维护 Web 应用程序变得更加容易。

12.1　Struts 简介

与 Tomcat 一样，Struts 也是 Apache 软件基金会的 Jakarta 项目的一部分，它是一个很成功的开放源代码项目。Struts 框架完全是用 Java 编写的，它使用了标准的 J2EE API。Struts 可用于维护和扩展 Web 应用程序。

12.1.1　Struts 概述

在使用 Java 技术建立的 Web 应用系统中，由于 JSP 技术的发展，在 JSP 页面中可以非常容易地结合业务逻辑(jsp:useBean)、服务端处理过程（jsp:scriplet）和 HTML(<html>)。在 JSP 页面中同时实现显示、业务逻辑和流程控制，从而可以快速地完成应用开发。因此 Web 应用系统可以仅由一组 JSP 页面构成，这种以 JSP 为中心的开发模型称之为 Model 1。

Model 1 开发模式仅适合于小规模的应用系统。Model 1 将业务逻辑和表示逻辑混合在 JSP 页面中，没有进行抽象和分离。在开发大型的 Web 应用时，非常不利于应用系统业务的复用和改动。如果要进行改动，必须在多个地方进行修改，这样非常不利于应用系统的扩展和更新。

考虑到 Model 1 的这些问题，在大型 Web 应用中必须采用不同的设计模式，把应用逻辑、处理过程和显示逻辑分成不同的组件实现，这些组件可以进行交互和复用，这就是 Model 2。

Model 2 具有组件化的优点，从而更易于实现对大规模系统的开发和管理，但是开发 MVC 系统比简单的 JSP 开发要复杂许多，因此需要一个很好的设计和开发框架，Struts 就是这样的一种框架。

Struts 使网页设计人员和 Java 开发人员能将精力集中于自己最擅长的方面。但是，在强壮性增强的同时，也意味着增加了复杂性。但对于更大的系统而言，Struts 则有助于管理复杂性。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Struts 框架包是 Model 2 思想的开放源代码实现。Struts 首先由 Craig R. McClanahan 创建，并于 2000 年捐献给 Apache 软件基金会。

12.1.2 Struts 的获取和安装

Struts 不仅在开放源代码社区得到广泛应用，而且许多重要的商业软件（如 IBM 的 WebSphere Studio 5.0 以上版本、JBuilder 8.0 以上版本）都支持 Struts。

Struts 已经成为了用 Java 创建 Web 应用的一个最流行的框架工具，Struts 所实现的 MVC 模式给 Web 应用带来了良好的层次划分，同时也提供了一系列的工具来简化 Web 应用的开发。基于 Struts 框架的优秀产品很多，如 Easy Struts、Struts Console 等，这些都能和 Eclipse 进行很好的结合。

Easy Struts 是开放源代码组织 sourceforge.net 上的一个项目，可以在它的网站 http://sourceforge.net/projects/easystruts 上下载压缩文件 org.easystruts. eclipse_0.6.4.zip。

将下载后的 zip 文件解压缩到 Eclipse 的安装目录下的\eclipse\plugins 子目录中，该压缩文件对应的解压缩文件是 com.cross.easystruts.eclipse_0.6.4 子目录。该过程即完成了 Easy Struts 在 Eclipse 中的安装。

从 Apache 网站下载 Struts，下载 Struts 的网址为：http://jakarta.apache.org/site/binindex.cgi，二进制文件可以有 zip 或 tar.gz 等形式，如在 Windows 中运行，则下载 zip 文件 jakarta-struts-1.1.zip，并将其解压缩到任一目录中，该目录为 struts 的安装目录，zip 文件的解压缩文件对应为该安装目录中的 jakarta-struts-1.1 子目录。

12.1.3 Struts 的配置

Easy Struts 安装完成后，即可在 Eclipse 中对 Struts 进行配置，具体步骤如下。

① 启动 Eclipse。

② 选择“窗口”→“首选项”（“Windows”→“Preferences”），打开设置窗口。

③ 在窗口左侧的“对象树”中选择“Easy Struts”（在安装 Easy Struts 之前，对象树中没有此对象），则在首选项窗口的右侧面板中，可以设置 Easy Struts，如图 12.1 所示。

④ 选择 Struts1.1 选项卡，如图 12.2 所示。

⑤ 在 Struts 选项卡中，单击“Add JARs...”按钮将 Struts 安装目录下的\jakarta-struts-1.1\lib 目录中的.jar 文件逐一添加进来。这些.jar 文件包括：

```
commons-beanutils.jar
commons-collections.jar
commons-digester.jar
commons-fileupload.jar
commons-lang.jar
commons-logging.jar
commons-validator.jar
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

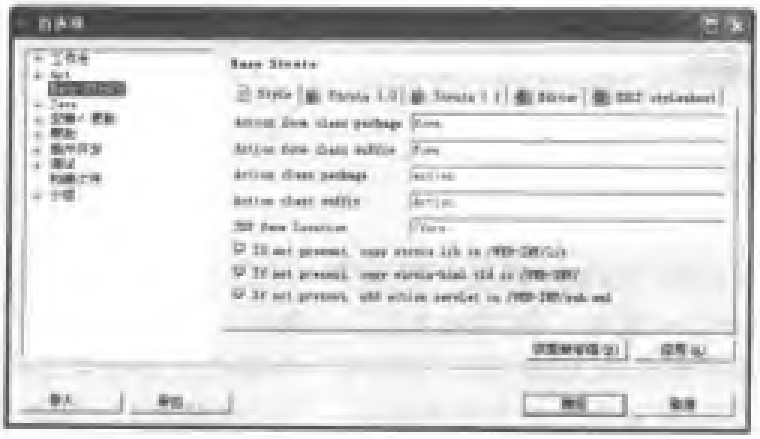


图 12.1　首选项——Easy Struts 的设置

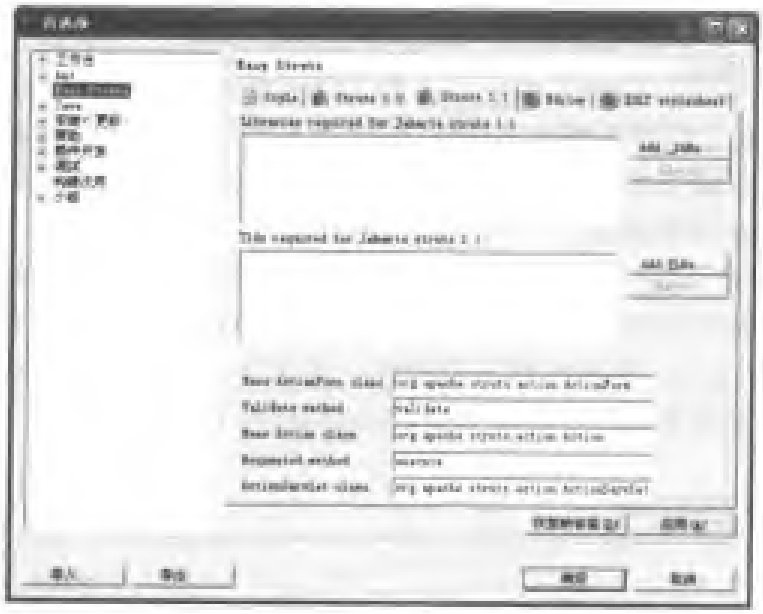


图 12.2　Struts1.1 选项卡

⑥ 在 Struts 选项卡中，单击“Add_TLDs...”按钮将 Struts 安装目录下的\jakarta-struts-1.1\lib 目录中的.tld 文件逐一添加进来。这些.tld 文件包括：

```
struts-bean.tld
struts-html.tld
struts-logic.tld
struts-nested.tld
struts-template.tld
struts-tiles.tld
```

⑦ 单击“确定”按钮，完成 Easy Struts 的设置。

12.2　Struts 框架概述

模式－视图－控制器（Model-View-Contriller）简称 MVC。Struts 是对 MVC 设计模

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

式的一种实现，所以 MVC 是 Struts 的基础。Struts 将 J2EE 规范中的 Servlet 2.2 和 JSP 1.1 规范用做实现的一部分。

12.2.1 MVC 模式

MVC 是一种设计模式，它是 J2EE 应用开发中的一个重要概念。MVC 设计模式主要用于代码复用，减少应用系统的开发周期。分层的方法在计算机科学中得到广泛的应用，如 OSI 七层协议、TCP/IP 协议等。应用分层是良好的软件设计的基础。在应用分层设计中，每一层都专门提供一类服务，并提供与其他层交互的一个接口。

为了生成健壮的、可维护的和可复用的应用程序，MVC 设计模式将应用程序组件分到三层中的某一层：模型（Model）、视图（View）或控制器（Controller），如图 12.3 所示。

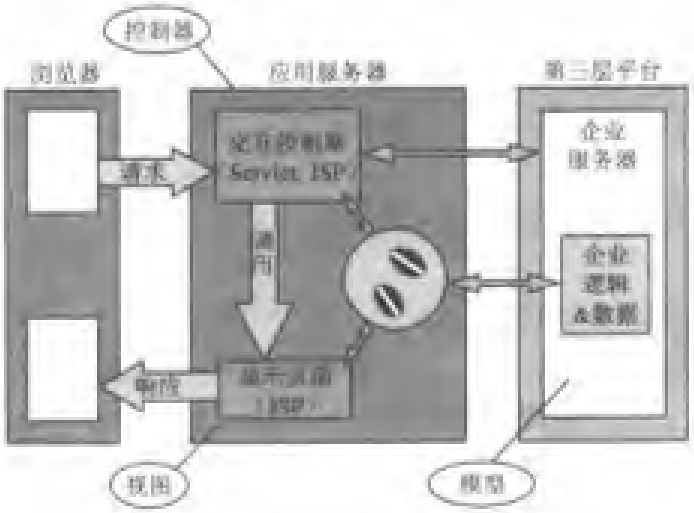


图 12.3 基于 MVC 体系结构的应用程序

通过区分各个层，MVC 允许组成每个层的各个组件间松散地耦合。这将使得程序开发更加灵活，并且可以重用代码。

模型层表示底层的数据和业务逻辑，模型层通常又可分成许多不同的子层。通常，模型充当现实世界中的过程的软件模拟，这样，在定义模型的时候将可以应用真实世界的建模技术了。在设计模型层组件时重要的一点是：必须保证模型层与用户界面（输入—输出方式）无关。在 J2EE 应用程序中，通常使用 JavaBeans 或 EJB 来封装业务逻辑和规则并且实现业务处理。

视图层为用户界面，主要用于表示。视图（表示）层包含接受用户输入以及表示应用程序输出的组件。视图通过模型访问企业数据，并指定应该如何表示该数据。在一个基于服务器端的 J2EE 应用程序中，典型的组件包含 HTML 表单、JavaServer Pages（JSP）页面，以及专门用于表示的 Servlet。

控制器层连接模型和视图。控制器将和视图之间的交互转换为由模型执行的操作。控制器层包含的组件通常用于解释输入请求、从请求中提取输入参数以及选择合适的模型组件来执行操作。控制器扮演交通警察的角色，当模型层一项操作完成后，控制器将所

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

有数据结果打包，并传送到合适的视图组件来传递应用程序的响应。控制器通常是 Servlet。

控制器也常常更新应用程序状态变量，因此控制器可以很容易地知道后续请求的应用程序状态。假设有一个商业类型的应用以及一个结账请求。这项操作有一个前提条件，应用中将要用到一个非空的购物车，以及潜在的其他相关对象（如发货地址和账单地址）。控制器需要很快地确定外部输入的用户结账请求是否可行（如：前提条件是否满足）。只有满足条件，控制器才会将请求传送到合适的业务逻辑（模型层）。如果发现前提条件没有满足，那么控制器将进入到一个相应的视图组件，重新建立客户端（浏览器）和应用程序之间的状态相容性。

MVC 设计模式有利于代码复用，业务处理模块往往无须进行额外的编码就可在不同的应用程序中重用。MVC 可以加速应用系统的开发，如模型层、视图层和控制器层可并行开发。由于 MVC 方法将应用系统分解为表示、逻辑和控制三部分，各部分的关联较小，从而提高了应用系统的可维护性。视图层表示的修改可以不影响模型层的业务逻辑，例如，在视图层的 Web 网页显示中可以使用图形显示取代原有的表格形式的显示，而无须修改模型层。类似地，模型层的修改也不会影响表示层，例如，银行的利率可能会有一些变动，而客户端的显示方式可以保持不变。

12.2.2 J2EE 中 MVC 模式的实现

为了加快企业应用程序的设计和开发，J2EE 平台提供了一个基于组件的方法来设计、开发、装配和部署企业应用程序。J2EE 平台提供了多层的分布式应用模型、组件重用、一致化的安全模型和灵活的事务控制。

J2EE 应用充分地利用了 MVC 的优越性。图 12.4 表示了 MVC 模式的 J2EE 实现框架中涉及的主要 J2EE 组件。

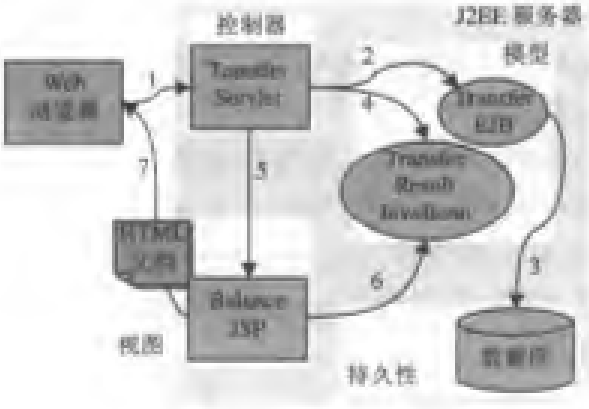


图 12.4 MVC 模式的 J2EE 实现

当用户通过单击一个超级链接或者在浏览器中提交一个表单和内容交互时，Servlet 将起着控制作用。Servlet 协调使用 JavaBean、EJB 或者两者一起执行业务任务、访问信息。Servlet 封装了一个或者多个 JavaBean 的结果，在没有传递到 JSP 之前，结果将被存储在一些上下文环境（通常是请求）中，以后由 JSP 来访问。JSP 检索包含结果值的

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

JavaBean，并对结果值进行格式化，以便于在浏览器中显示。

通过将模型与视图进行分离，信息的表示可以设计为可插入的和可替换的。所以同一业务对象在不同的浏览器、个人数字助理（PDA），或者其他网络设备中，可以具有各不相同的视图。

同样，当开发人员创建一个分离的模型类时，他们可以在其他的企业应用程序中复用这个模型类。复用是面向对象编程语言的最大好处，可以提高开发效率、降低开发开销。MVC 有助于实现这一点。

最后，因为命令逻辑和控制逻辑是分离的，控制器的职责将仅限于在用户、模型和视图间进行协调。由于限制了 Servlet 的职责，开发人员将可以创建“专而精”的组件。开发人员创建控制器时，不需要担心如何显示结果，只需考虑如何得到结果。

12.2.3 Struts 框架

许多 Web 应用程序有这种形式的工作流：其中客户执行一个调用一些控制器代码的请求，调用这些控制器代码可以确定下一步向客户显示什么，以及从那里可以采取什么操作。例如，客户可能发出登录请求，在失败时，用户将返回到登录页，成功时，用户将被转发到列出了他可以请求的后续服务页面。通常，如何请求一个操作、以及不同的结果引导到什么地方等工作流处理是直接嵌入在 Java 控制器代码中的。有了 Struts，不同的请求，也称为 action，可独立编码，从而可在高层的 XML 控制器文件中编写脚本控制工作流。这个控制器文件决定操作是如何调用的、结果会引导到什么地方及显示哪种表示（JSP）。如果有了结构良好的操作，基于 Struts 的应用程序是非常灵活的，对工作流的改变不需要重新编码，只需要改变控制器文件。

Struts 是一组相互协作的类、Servlet 和 JSP 标记，它们组成一个可重用的 MVC 设计。这个定义表示 Struts 是一个框架，它是 MVC 设计范例的一个变体。该框架包含一组 Java 类和 JSP 标记库。Struts 框架是对 MVC 设计模式的一种实现。Struts 是一个比较好的 MVC 框架，提供了对开发 MVC 系统的底层支持。同 Eclipse 工具一起，Struts 可以帮助开发人员快速地设计和实现 Web 应用程序。Struts 本质上就是使用 Servlet 和 Java Server Pages 技术的一种 MVC 实现，它将 J2EE 规范中的 Servlet2.2 和 JSP1.1 标记作为实现的一部分。图 12.5 显示了 Struts 的一个总的框架。

在图 12.5 中，来自客户浏览器的每个 HTTP 请求创建一个事件。Web 容器将用一个 HTTP 响应作出响应。控制器的作用是从客户端接受请求，并且选择执行相应的业务逻辑，然后把响应结果送回到客户端。

作为一个 MVC 的框架，Struts 对模型、视图和控制器都提供了对应的实现组件。Struts 框架的组件结构如图 12.6 所示。

如图 12.6 所示，在基于 Struts 的 Web 应用程序中，Web 浏览器所有的请求都提交给 ActionServlet 处理。ActionServlet 根据 struts-config.xml 文件中预先配置好的设置，将请求转发给相应的 Action 子类。若用户提交表单，则创建一个 ActionForm 子类，接受表单数据。Action 子类从 ActionForm 子类中获取数据，并调用业务逻辑代码进行处理，完

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

成业务处理功能。处理的结果值将递交给 JSP 页面，然后在用户的浏览器中显示。

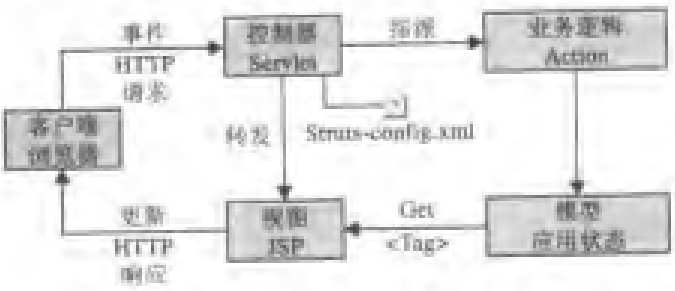


图 12.5　Struts 的一个总的框架

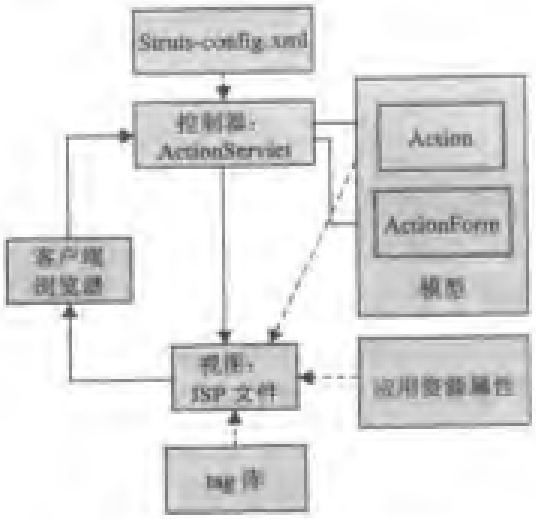


图 12.6　Struts 框架的组件结构图

就 Struts 而言，控制器是用 Servlet 实现的一个命令设计模式。struts-config.xml 文件配置控制器。业务逻辑更新模型的状态，并帮助控制应用程序的流程。就 Struts 而言，这是通过作为实际业务逻辑“瘦”包装的 Action 类完成的。

在 Struts 中，控制器的功能由 ActionServlet 和 ActionMapping 对象构成：核心是一个 Servlet 类型的对象 ActionServlet，它用来接受客户端的请求。ActionServlet 包括一组基于配置的 ActionMapping 对象，每个 ActionMapping 对象实现了一个请求到一个具体的模型部分中 Action 处理器对象之间的映射。

MVC 系统中的模型部分从概念上可以分为两类：系统的内部状态，改变系统状态的动作。Struts 为模型部分提供了 Action 对象。

所有的 Action 处理器对象都是开发者从 Struts 的 Action 类派生的子类。Action 处理器对象封装了具体的处理逻辑，调用业务逻辑模块，并且把响应提交到合适的视图组件以产生响应。

Struts 通常建议使用一组 JavaBean 表示系统的内部状态。根据系统的复杂度也可以使用实体 EJB 或会话 EJB 等组件来实现系统状态。Struts 建议在实现时把“做什么（Action）”和“如何做（业务逻辑）”分离。这样可以实现业务逻辑的重用。

ActionForm bean 在会话级或请求级表示模型的状态，而不是在持久级。JSP 文件使

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

用 JSP 标记读取来自 ActionForm bean 的信息。视图就是一个 JSP 文件，其中没有流程逻辑，没有业务逻辑，也没有模型信息，只有标记。标记是使 Struts 有别于其他框架的特点之一。

Struts 提供的 ActionForm 组件对象可以通过定义属性描述客户端表单数据。开发者可以从它派生子类对象，利用它和 Struts 提供的自定义标记库结合可以实现对客户端的表单数据的良好封装和支持，Action 处理器对象可以直接对它进行读写，而不再需要和 request、response 对象进行数据交互。通过 ActionForm 组件对象实现了对视图和模型之间交互的支持。

Struts 应用中的视图部分是通过 JSP 技术实现的。Struts 提供了自定义的标记库可以使用，通过这些自定义标记可以非常好地和系统的模型部分交互，通过使用这些自定义标记创建的 JSP 表单，可以实现和模型部分中的 ActionForm 的映射，完成对用户数据的封装，同时这些自定义标记还提供了如模板定制等多种显示功能。

综上所述，Struts 对应的 Web 应用处理流程为：Struts 控制器的 ActionServlet 处理客户请求，利用配置的 ActionMapping 对象把请求映射到 Action 处理器对象进行处理。Action 处理对象访问 ActionForm 中的数据，处理和响应客户请求，它还调用后台的 Bean 组件，这些组件封装了具体的业务逻辑。Action 处理器对象根据处理结果通知控制器，并通过 JSP 返回给用户。

Struts 是一个开发 web 应用程序的概念框架，而不是一个库，但 Struts 也包含了丰富的标记库和独立于该框架工作的实用程序类。Struts 框架包和库的概念还有微妙的差异。通常库包含了程序可以调用的函数或子程序。框架包则提供了通用的、相互协作的、可以复用的组件，Web 应用程序通过扩展这些组件提供特定的功能。

框架包最简单的形式是一组类和接口，它们相互协作以解决特定类型的软件问题。框架包包括大量类或组件，每一种类或组件都提供了某种概念的抽象。框架包定义了这些抽象如何协作解决问题，并在更高层次上组织模型。

Struts 框架包由大约 300 个 Java 类组成，分为 8 个顶级核心包，少数顶级包还包含子包。表 12.1 对每一种顶级包进行了简要描述。

表 12.1　Struts 框架包中的顶级包

包　名	描　述
action	包含控制器类 ActionForm、ActionMessage 以及其他少数需要的框架包组件
actions	包含现成的 Action 类，如 DispatchAction，可以为应用程序所用，或被应用程序扩展
config	包括在 Struts 配置文件中表示的配置类
taglib	包含用于 Struts 标记库的标记处理程序类
tiles	包含 Tiles 框架包使用的类
upload	包括用于通过使用浏览器从本地文件系统上载和下载文件的类
util	包含整个框架包使用的通用工具类
validator	包含当部署验证器时，Struts 框架包使用的针对 Struts 的扩展类。实际的 Validator 类和接口位于 common 包，这个包与 Struts 框架包相分离

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

12.2.4 基于 Struts 框架的开发流程

目前，Struts 已经是 Web 应用开发事实上的标准。根据已有的大量项目实践经验，基于 Struts 框架的 Web 应用项目的开发流程主要含以下步骤。

① 收集和定义应用需求。
② 基于数据采集和显示的原则定义和开发“屏幕显示”需求 。
③ 为每一个“屏幕显示”定义访问路径。
④ 定义 ActionMappings 建立到应用业务逻辑之间的联系。
⑤ 开发满足“屏幕显示”需求的所有支持对象。
⑥ 基于每一个“屏幕显示”需求提供的数据属性来创建对应的 ActionForm 对象。
⑦ 开发被 ActionMapping 调用的 Action 对象。
⑧ 开发应用业务逻辑对象，如 JavaBean、EJB 等。
⑨ 对应 ActionMapping 设计的流程创建 JSP 页面。
⑩ 建立合适的配置文件 struts-config.xml 和 web.xml。
⑪ 测试和部署。

在使用 Struts 框架时对应的各个部分的开发工作，我们将在后续章节中具体介绍。

12.3 视图及其构建

视图提供模型的表示，它是应用程序的外观。视图可以访问模型的读方法，但不能访问写方法。此外，视图无须了解控制器，它可以对控制器一无所知。当模型发生变化时，需要使用事件通知表单来通知视图。

12.3.1 视图简介

Struts 采用视图组件为客户端提供动态内容，主要由 JSP 页面完成视图。Struts 框架包中的 ActionForm 对象在用户和模型层间传递客户机输入数据。

为了使用 ActionForm 对象，必须用 Struts 提供的自定义标记创建 HTML 表单。利用 Struts 提供的自定义标记库编写用户界面把应用逻辑和显示逻辑分离。Struts 框架通过这些自定义标记建立了视图和模型之间的联系。Struts 的自定义标记还提供了很多定制页面的功能。

虽然 JSP 开发人员可以使用 scriptlet 标签和表达式标签在 JSP 中插入 Java 代码，但通常最好自定义标记，而不是 scriptlet 或表达式，原因在于：

- scriptlet 将逻辑与表示混在一起。
- scriptlet 破坏了常规的任务划分。
- scriptlet 使 JSP 页面更难于阅读和维护。

而自定义标记却不存在这些问题，并支持复用，因为同一个标记可以插入到多处，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

而其实现却在一个地方，从而有利于维护和更新。

12.3.2 在 Struts 中构建视图

浏览器发出的请求可带参数。当用户提交 HTML 表单后，Struts 框架将参数放在 org.apache.struts.action.ActionForm 的一个子类中。ActionForm 维护 Web 应用程序的会话状态。ActionForm 是一个抽象类，必须为每个输入表单模型创建该类的子类。

ActionForm 表示 HTTP 窗体中的数据，可以将其看做是模型和视图的中介，它负责保存视图中的数据供模型或者视图使用。

ActionForm 是一个抽象类，必须通过继承来实现自己的类。一旦创建了由 ActionForm 派生而来的类，就需要在 Struts 配置文件中配置它，需要将一个新的 form-bean 元素添加到 struts-config.xml 文件的 form-beans 部分，下面是在 struts-config.xml 文件中定义所实现的 ActionForm 派生类 LoginForm 的示例：

```
<form-beans>
  <form-bean
    name="loginForm"
    type="edu.sjtu.form.LoginForm"
  </form-bean>
</form-beans>
```

类型字段 type 的值必须为 ActionForm 派生而来的 Java 类名全称。

可将多个请求映射到用户的 ActionForm。struts-config.xml 文件控制 HTML 表单请求与 ActionForm 之间的映射关系。Struts 将使用 HttpServletRequest 中相应的域设置 ActionForm 的状态。在将它传递给 Action 类之前，Struts 还会调用 validation()方法进行表单状态验证。Struts 1.1 文档中把 ActionForm 比做 HTTP 和 Action 之间的防火墙，这体现了 ActionForm 具有的过滤保护作用，只有通过 ActionForm 验证的数据才能够发送到 Action 处理。

12.4 控制器及其构建

在基于 Stuts 的 Web 应用程序中，Servlet 提供了系统的单一入口点。客户端不直接向 JSP 页面发送请求，而由 Servlet 执行前端处理过程，包括身份验证和授权、单点登录控制，以及国际化设置等。

12.4.1 控制器简介

在独立的 GUI 客户端应用程序中，用户和控制器间的交互可能是按钮单击和菜单选择；在 Web 应用程序中，用户和控制器间的交互则可能是 GET 和 POST HTTP 请求。控制器根据用户交互和模型操作的结果选择合适的视图，从而作出响应。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

GET 方法从服务器请求一个由 URI 指定的资源。当该资源是一个类似于 Servlet 的产生数据的资源时，产生的数据将由相应消息返回，并由客户端进行处理。如果所请求的资源不存在，则在响应中返回错误头，指示一个 404（没有找到）错误。

对于 Web 应用程序，POST 方法用于调用服务器上与该请求相关的资源。这些资源通常处理比仅返回一个文档更复杂的请求。

在 Action 和 ActionForm 一样都可以实现验证方法，那么如何来安排它们之间的分工呢？一般来说，按照 MVC 的原则，视图级的验证工作放在 ActionForm 来完成，比如输入不能为空，E-mail 格式是否正确，利用 ValidatorForm 可以很轻松地完成这些工作。而与具体业务相关的验证则放入 Action 中，这样就可以获得最大 ActionForm 重用性的可能。

12.4.2 在 Struts 中构建控制器

ActionServlet 被实现为一个专门处理 HTTP 请求的 Servlet，它同时具有 Servlet 的特点。ActionServlet 的类型是 org.apache.struts.action.ActionServlet，它负责处理所有的浏览器请求，但并不进行任何具体的处理，而是使用 struts-config.xml 文件将请求映射到相应的 Struts 的 Action 子类。使用这种方法有以下几个优点。

- 应用程序的整个逻辑流程都存储在一个分层的文本文件中。这使得人们更容易查看和理解它，尤其是对于大型应用程序而言。
- 网页设计人员不必费力地阅读 Java 代码来理解应用程序的流程。
- Java 开发人员也不必在更改流程以后重新编译代码。

在 Struts 中，ActionServlet 主要完成以下功能。

- 接收客户端请求。
- 根据客户端的 URI 将请求映射到一个相应的 Action 类。
- 从请求中获取数据填充 Form Bean（如果需要）。
- 调用 Action 类的 execute()方法获取数据或者执行业务逻辑，如存取数据库，调用实体 bean 等。
- 选择正确的视图响应客户端的请求。

控制器通过一组动作映射来决定所调用的动作实例。动作映射是 Struts 配置信息的组成部分。每一个 action 元素都在内存中由 org.apache.struts.action.ActionMapping 类的一个实例代表。Struts 提供了核心控制部分的实现，开发者只需要配置 ActionMapping 对象。通过配置<action-mappings>实现了从请求到相应的 Action 的映射。下面是 ActionMapping 配置的一个示例：

```
<action-mappings>
  <action
      name="loginActionForm"
      path="/loginAction"
      type="edu.sjtu.StrutsSample.action.LoginAction"
      input="/login.jsp"
      scope="request"
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理


```
        <forward name="failure" path="/login.jsp"/>
        <forward name="success" path="/index.jsp"/>
    </action>
</action-mappings >
```

ActionMapping 对象包含一个 path 属性，此属性将与客户端请求的 URI 进行匹配。当与来自客户端的请求匹配时，则将请求映射到相应的 Action 类。ActionServlet 调用 Action 的 execute()方法。如果 Action 类不存在，控制器就调用 Action 类的一个实例。

Action 类是应用程序流程的一个封装。Action 类的用途是将 HttpServletRequest 转换为对业务逻辑的调用。要使用 Action，需创建它的子类并覆盖 execute()方法。方法 execute 的签名为：

```
public ActionForward execute (ActionMapping mapping,
                              ActionForm form,
                              HttpServletRequest request,
                              HttpServletResponse response)
    throws Exception;
```

通过继承 Action 类可实现具体的执行类。具体 Action 类的功能一般都在 execute()（以前是 perform 方法）方法中完成。

方法 execute()的返回类型是 org.apache.struts.action.ActionForward。当动作完成后，Struts 框架包还使用映射信息决定将用户转发到哪一种资源。如前面 ActionMapping 配置的示例所示：与动作映射一样，动作转发信息也在配置文件中规定，并定义在 Action 级。每一条转发配置必须包含 name 和 path 属性，此外还包含一个可选属性 redirect。如果不指定 redirect 属性，其默认值就是“false”。

Action 类应该控制应用程序的流程，而不应该控制应用程序的逻辑，因为将业务逻辑放在单独的包或 EJB 中，可以提供更大的灵活性和可重用性。

关于 Action 类属于控制器层还是模型层有一些争议，但通常将 org.apache.struts.action.Action 类看做控制器组件的扩展。

12.5 模型及其构建

模型包含应用程序的核心功能，如业务实体以及控制访问和修改数据的规则。模型封装了应用程序的状态。有时它包含的惟一功能就是状态。模型应该保持对用于访问业务对象及其相关规则的客户端类型的无关性。模型无须了解视图或控制器，可以对视图或控制器一无所知。换句话说，模型层组件不应该意识到是何种类型的客户端或框架在使用它。

应该尽量使业务逻辑置于 Action 类之外，Action 只负责错误处理和流程控制，而将业务逻辑执行分离到单独的 JavaBean 中。

Struts 的模型部分通常采用 JavaBean 和 EJB 组件，设计和实现系统的业务逻辑，但 Struts 框架并没有将开发者限制到某一个具体的实现模型上，Struts 框架支持不同的组件模型用来处理应用的业务域。根据不同的请求从 Action 派生具体 Action 处理对象。完成

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

“做什么”的任务来调用由 Bean 构成的业务组件。创建由 ActionForm 的派生类实现对客户端表单数据的封装。

J2EE1.3 规范直接指出 EJB“通常包含了 J2EE 应用程序的业务逻辑”。EJB2.0 定义了三种不同类型的 EJB 的体系结构和行为。在 EJB 的三种类型中，会话 bean 提供事务性的、分布式的和可管理的资源组件，但并不支持任何持久性。这些特性并不表示会话 bean 没有状态，只是容器不会保持会话 bean 的状态。另外，会话 bean 的生命周期并不表明它有可能改变状态。因此，会话 bean 在 Web 应用程序开发中起到特殊的作用，可以将会话 bean 用做任务处理器或者流程控制器。在 EJB 的所有应用程序中，会话 bean 通常作为处理途径或者处理程序，来实现应用的模型。其他 EJB 类型更适合表示或者管理对象的状态。实体 bean 用来表示 J2EE 应用程序的持久性特征。它们支持开发者创建一些具有持久性结构的面向对象的视图。通常，由关系数据库管理持久性。

由模型执行的操作包括激活业务流程或改变模型状态。实体 EJB 不仅可用于执行业务逻辑，而且用于持久性管理。实体 bean 可用来添加一个持久性抽象层，并给应用程序提供一个更合理的接口。应用程序的其他组件可以使用这一层提供的持久性服务。实体 bean 将和普通的数据库操作紧密关联。

消息驱动 bean 是 EJB 家族的一个新成员。它允许应用程序开发者把 HTTP 驱动的应用程序（以及它的逻辑）从请求—响应模式中分离。消息驱动 bean 是 Java 消息服务（Java Message Service，JMS）的消息使用者，因此客户从不会直接使用它。相反，消息驱动 bean 与应用程序逻辑异步工作，来处理发送到特定目标的消息。

12.6 Struts 应用的配置

12.6.1 国际化 Struts 应用程序

Web 应用程序通常运行在 Internet、Intranet 或 Extranet 上，可能有各类形形色色的人访问，甚至有可能被全世界的人访问，因此开发和部署 Web 应用程序时，不仅需要解决应用逻辑的处理问题，而且还应该考虑如何使应用程序国际化。国际化通常称为“i18n”，因为国际化的单词“internationalization”以字母 i 开头，以 n 结尾，中间包含 18 个字母。

国际化的 Web 应用程序支持用户选择显示信息的语言，然后 Web 应用程序根据选定的语言加载应用程序的文本。对被支持的每种语言而言，应用程序的文本信息都存储在一个外部文件中，并且与另外一个文件的关键词相对应。应用程序代码使用这些关键词以及选定的语言来加载合适的文本。国际化 API 还提供类来根据选定的语言格式化日期和金钱。

Java 本身就支持多国语言编码。Java 国际化类库中最重要的类是 java.util.Locale。当创建一个 Locale 对象时，一般会指定语言及国家代码，如下列代码创建一个 US 区域：

```
Locale usLocale = new Locale ("en","US");
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

Struts 框架提供的国际化支持主要体现在应用程序中的文本以及各类消息的表示上。尽量不要在 JPS 文件中输入中文，而放在 properties 文件中。所有 JSP 用 UTF-8 编码：

```
< %@ page contentType="text/html;charset=UTF-8" % >
```

针对不同的语言，编写不同的 application.properties，例如英文资源文件 ApplicationResources.properties()和中文资源文件 ApplicationResources_zh_CN. properties。然后对这些配置信息文件用 native2ascii 工具处理一次，native2ascii 命令格式如下：

```
native2ascii  -encoding  gb2312  ApplicationResources_old.properties
ApplicationResources_zh.properties
```

通过 struts-config.xml 文件中的<message-resources>元素，可以指定包含应用程序的本地化消息的消息资源绑定。每一个 Struts 可以有一个或多个消息资源绑定。

12.6.2 web.xml 文件

Seb 应用中的 web.xml 是第一个要配置的地方。web.xml 定义了 struts 的控制类、config 文件和标签。下面是 web.xml 文件代码的一个示例：

```
<?xml version="1.0"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>
  <!-- Action Servlet Configuration -->
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <!-- Action Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <!-- The Welcome File List -->
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <!--  Struts Tag Library Descriptors -->
  <taglib>
    <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
  </taglib>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    <taglib>
      <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
    </taglib>

    <taglib>
      <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
    </taglib>
  </web app>
```

在上面的示例中，<servlet>部分定义了 Struts 提供的控制器，并可以为它指定初始化参数；<servlet-mapping>定义了客户请求的 URI 信息和服务器端具体处理的映射关系；<taglib>部分添加了在 Web 应用中所使用的自定义标记库的引用。

12.6.3 struts-config.xml 文件

struts-config.xml 是用于建立控制器和模型之间的关系的。它描述了控制器所使用的把请求对应到具体处理的法则，同时它还描述了客户提供的数据与 ActionForm 组件的对应映射关系。

在前面的几个小节中我们已经讨论了 struts-config.xml 文件中的一些重要元素，如<form-beans>、<action-mappings>和<message-resources>等。下面将简要讨论<data-sources>和<global-forwards>元素。

1. <data-sources>元素

理想情况下，业务逻辑应封装所有的数据访问细节，包括数据库连接的获得。但是，应用程序开发人员有时需要从 DataSource 对象来获得数据库连接。在这种情况下，可以通过在 struts-config.xml 文件中定义<data-sources>元素，分发和配置任何实现了 javax.sql.DataSource 接口的数据库连接池。在＜data-sources＞元素中可以定义多个数据源。使用<data-sources>元素配置数据源的代码示例如下：

```
<data-sources>
  <data-source>
    <set-property property=" id"  value="DS1" />
    <set-property property="key"  value="conPool" />
    <set-property property="description"  value="MySql Data Source" />
    <set-property property="driverClass"  value="edu.sjtu.StrutsSample.
    mysql.Driver" />
    <set-property property=" maxCount"  value="8" />
    <set-property property=" minCount"  value= "2"/>
    <set-property property="user"  value="admin" />
    <set-property property="password"  value="admin123" />
    <set-property property="url"  value="jdbc:mysql://localhost/test" />
    <set-property property=""  value= />
    <set-property property=""  value= />
  </data-source>
</data-sources>
```

示例代码中的主要属性的含义如下：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

id：数据源的标识符。
key：servlet context 属性，该数据源存储在其中。
description：数据源的描述。
autoCommit：数据源创建的连接所使用的默认自动更新数据库模式。
driverClass：数据源所使用的类，用来显示 JDBC 驱动程序接口。
loginTimeout：数据库登录时间的限制，以秒为单位。
maxCount：最多能建立的连接数目。
minCount：要创建的最少连接数目。
password：数据库访问的密码。
user：访问数据库的用户名。
url：JDBC 的 URL。
通过“key”，Action 类可以访问数据源，如：

```
javax.sql.DataSource ds = servlet.findDataSource("conPool");
javax.sql.Connection con = ds.getConnection();
```

2. <global-forwards>元素

如果有多个 Action 都可能返回同一个 forward，那么就可以将其定义为全局转发（global forward）。全局转发用来在 JSP 页之间创建逻辑名称映射。转发都可以通过对调用操作映射的实例来获得，下面是一个全局转发的示例：

```
<global-forwards>
   <forward name="personCheck" path="/CheckView.jsp"/>
</global-forwards>
```

在上面的代码中，name 表示全局转发的名字；path 表示与目标 URL 的相对路径。

12.7 Eclipse 中开发 Struts 应用的简单示例*

在 Eclipse 中安装并配置 Easy Struts 插件后，将可开发 Struts 应用。与开发其他 Web 应用程序类似，在 Eclipse 中开发 Struts 应用程序也需要创建一个项目，以便将包和类与 Web 项目关联起来，然后在项目中添加 Struts 支持，紧接着就可以开发 Struts 应用程序。下面介绍在 Eclipse 中创建 Struts 应用的过程。

12.7.1 创建 Struts 项目

首先在 Eclipse 中创建 Tomcat 项目，并添加 Struts 支持，其步骤如下。
① 选择“文件”→“新建项目”，则打开“新建项目”窗口，如图 12.7 所示。

* 本节内容参考文献 25。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 12.7　新建项目窗口

如图 12.7 所示，在新建项目窗口中单击“Java”，然后选择“Tomcat Project”。

② 单击“下一步”按钮，则打开“New Tomcat Project”窗口，如图 12.8 所示。



图 12.8　新建 Tomcat 项目

在新建 Java 项目窗口中，指定项目名，在本例中键入项目名 StrutsSample。

③ 单击“完成”按钮，完成 StrutsSample 项目的创建。

④ 在 Java 透视图的包资源管理器中选中所创建的项目 StrutsSample，并再次打开“新建”窗口（在菜单中选择“文件”→“新建”→“其他”；或右击项目名，从上下文弹出式菜单中选择“新建”→“其他”），如图 12.9 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 12.9　在所选择的项目上添加 Easy Struts 支持

在“新建”窗口中，在左边的窗格中选择“Java”下的子选项“Easy Struts”，并在右边的窗格中选中“Add Easy Struts support”。

⑤ 单击“下一步”按钮，进入 Easy Struts 项目的新建界面，如图 12.10 所示。



图 12.10　新建 Struts 支持

在“新建”窗口中，输入 Struts 支持的基本信息。在 Webapp folder 文本框中，可以指定 Web 应用程序的文件名。部署该项目时仅需将该目录打包为 WAR 文件即可。除在“Base package for generation”文本框中指定 Easy Struts 自动生成的程序包名，其他选项可接受默认值。

⑥ 单击“完成”按钮，则 Easy Struts 自动生成指定的程序包，并将一些.tld 文件拷

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

贝到 Struts 项目的 WEB-INF 子目录下，将一些.jar 文件自动拷贝到 WEB-INF\lib 子目录下，同时将 Struts 项目所需的包（一些.jar 文件）自动拷贝到 Struts 项目中，如图 12.11 所示。



图 12.11　Easy Struts 在项目中自动创建的包和文件

在包资源管理器中展开包文件夹，将可看到已自动创建的应用程序资源文件：ApplicationResources.properties，这个文件可以用来存放应用程序的资源信息。双击该文件，将会在属性文件编辑器中打开该文件，可以发现它现在尚是一空白文件，如图 12.12 所示。



图 12.12　ApplicationResources.properties 文件

在项目的 WEB-INF 目录中，这时已自动创建了 web.xml 文件，如图 12.13 所示。该文件已经自动定义好 ActionServlet 入口，以及这个 Servlet 所需的参数，开发者无须再行配置。

在开发具体的 Struts 应用程序之前，首先确认项目的类路径中是否已经添加了 struts.jar 以及 servlet.jar 这两个文件，否则新建 struts 程序时将会出现若干错误信息。并确认 WEB-INF\lib 目录下是否已有 struts.jar 文件，否则须手工将 struts.jar 文件拷贝到 WEB-INF\lib 目录下。完成这些工作后，将可正式进行 Struts 应用程序的开发。

在执行完以上准备工作后，即可进行 Struts 应用程序开发。选择“文件”→“新建”

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

→“其他”，将打开“新建”窗口，在“新建”窗口中选择 Java 的子选项 Easy Struts，即可选择创建相关的 Struts 应用程序对象。Easy Struts 插件已经在 Eclipse 开发环境中添加了几个用于创建 Struts 相关程序的向导。这些向导包括：

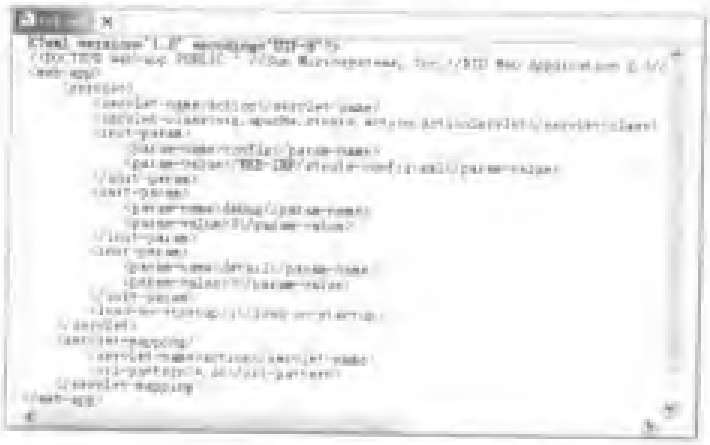


图 12.13　web.xml 文件

- Easy Form。创建一个具有表单属性的 JSP 文件，以及一个具有表单属性的 getter 方法和 setter 方法的 Form bean，并向配置文件中添加一个 Form bean 的定义。
- Easy Action。创建一个 Action 类，并向配置文件中添加一个操作映射的定义。
- Easy Action associated with a form。创建一个具有表单属性的 JSP 文件，一个具有 getter 方法和 setter 方法的 Form bean 类，以及一个 Action 类；并向配置文件中添加一个 Form bean 的定义和一个操作映射的定义。
- Easy Forward。创建局部转发和全局转发，它们定义了控制应该转向何处。
- Easy Exception。处理异常。
- Easy Message resources。创建资源属性文件，这对于内容的国际化和本地化特别重要。
- Easy Plug-in。创建插件。
- Easy Datasource。将应用程序连接到一个数据源上。
- Easy Module。模块化应用程序。

下面将使用“Easy Action associated with a form”向导和“Easy Forward”向导来构建一个简单却又完整的 Struts 应用程序。

12.7.2　使用“Easy Action Associated With a Form”向导

使用“Easy Action Associated With a Form”向导可创建一个具有表单属性的 JSP 文件，具有 getter 方法和 setter 方法的 Form bean 类，以及一个 Action 类；并可向配置文件中添加 Form bean 的定义和操作映射的定义。其步骤如下。

① 在项目的上下文菜单中选择“新建”→“其他”，打开“新建”对话框窗口。

② 在“新建”对话框窗口的左侧窗格中选中“Easy Struts”，然后在右侧窗格中选择“Easy Action Associated With a Form”。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

③ 单击“下一步”按钮，这将打开“Struts Form”的“New”对话框窗口，如图 12.14 所示。



图 12.14 新建 Struts Form

④ 在“New”对话框中，在“Use case”字段中输入用例名，本例中假设为“test”，向导将会根据用例名在“Form name”和“Form type”字段中自动生成表单名和表单类型。

⑤ 单击“Add”按钮添加表单属性，这将打开如图 12.15 所示窗口。



图 12.15 添加表单属性

在窗口中可添加表单属性。例如，添加一个文本输入域：在“Name”字段中输入名字“tel”。需要注意的是表单属性名不能以大写字母开头，否则系统就会警告在调用操作时找不到 getter 方法，并且表单属性名不能重复。在“Type”字段中输入类型“int”；在“Initial value”字段中将“Tel”的初值设置为 0。在设置“Type”时，可以直接输入，也可以单击“Browse”按钮选择。对于“JSP input type”的设置，既可以直接输入，也可以从下拉列表中选择，可供选择的类型有“text”、“textarea”、“password”、“checkbox”和“select”。完成输入后单击“确定”按钮将生成“Tel”表单域。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

然后依次生成名为“greet”、类型为“String”、初始值为 Mr.、JSP 输入类型为 select 的字段；名为“name”、类型为“String”、没有初始值、JSP 输入类型为 text 的字段；名为“email”、类型为“String”、没有初始值、JSP 输入类型为 text 的字段；名为“address”、类型为“String”、没有初始值、JSP 输入类型为 text 的字段。完成所有的表单域设置后，单击“完成”按钮，将回到“New”窗口。

Struts 系统会在 Form bean 类中为每个表单属性生成一对 getter 方法和 setter 方法。因此，对于一个选择列表来说，应该只创建一个<html:select>即可，而不用创建所有的<html:option>。

⑥ 由于开发的是 Web 应用程序，因此选中“public ActionErrors validate (HttpServlet Request...)”复选框和“public void reset (HttpServletRequest...)”复选框（它们是默认选择）。

⑦ 单击“下一步”按钮，这将进入配置关联 action 和 Struts form 的设置窗口，如图 12.16 所示。



图 12.16 关联 action 与已有的 Struts form

⑧ 在窗口中，“(Create JSP) input”字段的默认值是“/form/test.jsp”这意味着 JSP 文件被放在项目的“/form”文件夹中。开发者可以手工修改 JSP 文件的位置。操作映射的其他字段的含义如下。

- Path：所提交请求的相对路径，路径名必须以“/”开头而且惟一。
- Type：这个 ActionMapping 正在描述的 Action 类的名字。
- Attribute：可以访问这个 Form bean 的请求范围或会话范围的属性名，条件是名字与这个 bean 指定的名字不同。
- Scope：规定与这个映射相关的 Form bean 的值应该可以保存多少位。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

- input：规定如果碰到验证错误时控制流程应该返回到输入表单的相对路径。
- Validate：如果这个属性被设置为 true，就会对与这个映射关联的表单调用 ActionForm.validate() 方法。
- Parameter：可用于向由这个映射选择的 Action 传递其他信息。

⑨ 接受其他各项的默认值，并单击“下一步”按钮，将打开新的设置窗口，如图 12.17 所示。



图 12.17 操作映射属性

图 12.17 中所显示的是操作映射属性，其含义如下。

- Forward：指定要处理这个请求的 Servlet 或 JSP 资源的相对路径。ActionMapping 可以使用 findForward()将控制流程转换到这个 Servlet 或 JSP 资源上。
- Exception：与这个映射相关的异常处理程序。

⑩ 单击“Forwards”窗格旁的“Add”按钮，将打开转发的配置窗口，如图 12.18 所示。

在该窗口中，name 是惟一的标识符，在操作映射的 findForward()方法中可以用它来返回这个路径。如果 redirect 为 true，那么控制流程就会被转到一个重定向的页中，而不是一个转发的页中。这意味着创建一个新请求。如果 contextRelative 的值被设置为 true，那么路径就被认为是相对于整个 Web 应用的相对路径。如果该值被设置为 false，那么路径就被认为是相对于一个模块化应用程序模块的相对路径。

⑪ 在本例中，在“Forward name”字段中输入“success”；在“Forward path”字段中输入“/success.jsp”。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 12.18 转发配置

⑫ 依次在窗口中单击“确定”和“完成”按钮，这将返回“操作映射属性”窗口。

⑬ 单击“完成”按钮。

此时，系统将会在 StrusSample 的 WEB-INF/src/目录中的 sjtu.edu.struts.action 包中自动生成 TestAction.java 文件，并在同一目录的 sjtu.edu.struts.action 包中自动生成 TestForm.java 文件。同时还会在/form 目录下自动生成 test.jsp 文件。下面，将对这三个文件进行一些修改，从而完成整个表单，添加用户输入的验证方法并处理验证错误。

12.7.3 修改 ActionForm 类

打开 ActionForm.java 文件，然后编写 validate()方法的代码，对用户输入的表单属性进行验证。validate()方法的完整代码如下所示：

```
package sjtu.edu.struts.form;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

/**
 * TestForm.java created by EasyStruts - XsltGen.
 * http://easystruts.sf.net
 * created on 09-04-2004
 *
 * XDoclet definition:
 * @struts:form name="testForm"
 */
public class TestForm extends ActionForm {

    // --------------------------------------------------------- Instance Variables

    /** Email property */
    private String email;

    /** Greet property */
    private String greet = "Mr.";

    /** Address property */
    private String address;

    /** Tel property */
    private int tel = 0;

    /** Name property */
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    private String name;

    // ----------------------------------------------------------- Methods

    /**
     * Method validate
     * @param ActionMapping mapping
     * @param HttpServletRequest request
     * @return ActionErrors
     */
    public ActionErrors validate(
        ActionMapping mapping,
        HttpServletRequest request) {

            ActionErrors errors = new ActionErrors();
                if (greet == null || greet.trim().equals("")) {
                    errors.add("greet", new ActionError("error.greet"));
                }
                if (name == null || name.trim().equals("")) {
                    errors.add("name", new ActionError("error.name"));
                }
                if (address == null || address.trim().equals("")) {
                    errors.add("address", new ActionError ("error.
                    address"));
                }
                if (email == null || email.trim().equals("")) {
                    errors.add("email", new ActionError("error.noEmail"));
                }
                else if (email.indexOf("@")==-1) {
                    errors.add("email", new ActionError("error.
                    wrongEmail"));
                }
                if (tel==0) {
                    errors.add("tel", new ActionError("error.tel"));
                }
                return errors;

    }

    /**
     * Method reset
     * @param ActionMapping mapping
     * @param HttpServletRequest request
     */
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        email = "";
        greet = "Mr.";
        address = "";
        tel = 0;
        name = "";

    }

    /**
     * Returns the Email.
     * @return String
     */
    public String getEmail() {
        return email;
    }
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
/**
 * Set the Email.
 * @param Email The Email to set
 */
public void setEmail(String email) {
    this.email = email;
}

/**
 * Returns the Greet.
 * @return String
 */
public String getGreet() {
    return greet;
}

/**
 * Set the Greet.
 * @param Greet The Greet to set
 */
public void setGreet(String greet) {
    this.greet = greet;
}

/**
 * Returns the Address.
 * @return String
 */
public String getAddress() {
    return address;
}

/**
 * Set the Address.
 * @param Address The Address to set
 */
public void setAddress(String address) {
    this.address = address;
}

/**
 * Returns the Tel.
 * @return Int
 */
public int getTel() {
    return tel;
}

/**
 * Set the Tel.
 * @param Tel The Tel to set
 */
public void setTel(int tel) {
    this.tel = tel;
}

/**
 * Returns the Name.
 * @return String
 */
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
    public String getName() {
        return name;
    }

    /**
     * Set the Name.
     * @param Name The Name to set
     */
    public void setName(String name) {
        this.name = name;
    }

}
```

ActionError（"error.noEmail"）中的"error.noEmail"是资源属性文件中定义的一个关键字，它等价于"Enter your email"。在资源属性文件中，可以使用<html>代码。ActionErrors 中 add(java.lang.String property, ActionError error)方法的参数 property 是一个表单属性名。这个 ActionError 只能与一个指定的表单属性相关。例如，如果 E-mail 地址为空，或者只包含空格字符，那么在对表单进行验证时就会显示"Enter your email"消息。如果在 E-mail 地址中没有"@"字符，那么就会显示"Correct your email"消息，这是在资源属性文件中使用"error.wrongEmail"关键字定义的。这两个错误都只与表单属性"email"有关。

在资源属性文件 ApplicationResources.properties 中可以使用<html>代码。资源属性文件的内容是一些键—值对，如下所示：

```
errors.header=<h4>Validation Error(s)</h4><ul>
errors.footer=</ul><hr>

error.greet=<li>Choose your greet
error.name=<li>Enter your name
error.address=<li>Enter your address
error.tel=<li>Enter your contact number
error.wrongEmail=<li>Correct your email
error.noEmail=<li>Enter your email
```

在源代码文件中可以调用这些关键字。在每个资源属性文件中系统都需要 errors.header 和 errors.footer 这两个关键字。

12.7.4 修改 Action 类

修改 TestAction 类的 execute()方法来实现用户的逻辑，每个 Action 类都必须实现 execute()方法，它会将控制流程转向 success.jsp。

```
package sjtu.edu.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理


```
import org.apache.struts.action.ActionMapping;

import sjtu.edu.struts.form.TestForm;

/**
 * TestAction.java created by EasyStruts - XsltGen.
 * http://easystruts.sf.net
 * created on 09-04-2004
 *
 * XDoclet definition:
 * @struts:action path="/test" name="testForm" input="/form/test.jsp"
   validate="true"
 * @struts:action-forward name="/success.jsp" path="/success.jsp"
 */
public class TestAction extends Action {

    // ----------------------------------------------- Instance Variables

    // --------------------------------------------------------- Methods

    /**
     * Method execute
     * @param ActionMapping mapping
     * @param ActionForm form
     * @param HttpServletRequest request
     * @param HttpServletResponse response
     * @return ActionForward
     * @throws Exception
     */
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        TestForm testForm = (TestForm) form;
        String greet = testForm.getGreet();
        String name = testForm.getName();
        request.setAttribute("name", name);
        request.setAttribute("greet", greet);

        // Forward control to the specified success target

        return (mapping.findForward("success"));

    }

}
```

12.7.5 修改 JSP 文件

首先，为<html:select>添加<html:option>，完成表单，然后在表单后面添加一个<html:errors />，这用来演示一个与属性为“proerty”的<html:error>不同的行为。例如，<html:errors property="name"> 只会反映与表单属性“name”相关的用户输入的验证。可以使用 ActionErrors 类的 add(String property, ActionError error)方法为一个表单属性关联一些错误。<html:errors/>则会反映这个表单中所有的验证错误。这些错误会在调用

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

<html:error>的地方显示。test.jsp 代码如下：

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
  <head>
    <meta
      name = "Generator"
      content = "Easy Struts Xslt generator for Eclipse
(http://easystruts.sf.net)."
    <title>Struts Form for ownerForm</title>
  </head>
  <body>
    <html:form action="/owner">
      greet : <html:select property="greet">
                <html:option value=""></html:option>
                <html:option value="Mr.">Mr.</html:option>
                <html:option value="Miss">Miss</html:option>
                <html:option value="Mrs.">Mrs.</html:option>
              </html:select><html:errors property="greet"/>
      name : <html:text property="name"/><html:errors property="name"/></br>
      address : <html:text property="address"/><html:errors property="address"
      /> </br>
        email : <html:text property="email"/><html:errors property="email"
        /></br>
        tel : <html:text property="tel"/><html:errors property="tel"/></br>
        <html:submit/><html:cancel/>
    </html:form>
    <html:errors />
  <body>
</html>
```

success.jsp 的代码如下所示：

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<logic:present name="name" scope="request">
Thank you,
  <logic:present name="greet" scope="request">
    <bean:write name="greet" scope="request"/>
  </logic:present>
  <bean:write name="name" scope="request"/>
  </logic:present>
```

<logic:present>是一个 Struts logic 标签。当逻辑标签具有“name”属性时，就应该检查是否存在这个指定名的 JSP bean。关于<logic:present>的属性完整列表和 Struts logic 标签的完整列表，请参阅 Struts 用户指南。

<bean:write>是一个 Struts bean 标签。它被用来将指定的 bean 属性的值传递给当前的 JspWriter。当 bean 标签具有“name”和“property”属性时，就是指其属性被访问以获取由“property”所指定的值的 bean 的属性名。如果没有指定“property”属性，那么就会传递这个 bean 本身的值。

至此，完成该 Web 应用的构建。进一步的示例和相关示例可参阅参考文献 25。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 13 章　Eclipse 的版本控制及 CVS

由于使用 Eclipse 进行的大型开发通常需要持续开发和改进，常因发现错误和为增强功能需要对它进行修改。因此需要在一个很长的时期维护大量文件。如果用户只保存文件的最新版本，就很容易丢失版本的变化情况，不能恢复到最近版本前的状态或版本。当需要回溯到一个程序的老版本，如程序只有少量的版本，对程序的所有版本都进行了简单的备份，还是可行的。如程序包含大量的文件，或文件可能涉及多个版本时，这种简单备份的方法会造成版本的不一致和混乱的情况。Eclipse 的版本管理分为本地版本管理和团队版本管理两类。

13.1　Eclipse 的本地版本控制

Eclipse 内置了本地历史记录（Local History）的版本支持。即使没有配置任何源代码控制管理程序，Eclipse 仍然可以跟踪工作台中源代码和其他文件的修改。

简单的源代码控制机制称为本地历史（Local History）。工作台可以在本地历史的功能中选择“比较（Compare with）”、“替换为（Replace with）”和“从本地历史中恢复（Restore from Local History）”。比较和替换功能允许查看文件差别，在替换功能的情况下，可以回退到文件的不同版本。恢复功能允许恢复以前删除的文件中的元素，例如方法或变量定义。这些功能可从工程浏览器和轮廓视图中的文件的上下文菜单找到，如图 13.1 所示。



图 13.1　本地历史的功能

文件的本地编辑历史是在用户创建或修改文件时维护的。每当用户编辑和保存文件时，就会保存一个副本，以便可以将当前文件替换为先前编辑或者甚至恢复已删除的文件。在 Eclipse 中，每一次被保存的更改都可以得到恢复，而且可以精确到每一个方法的版本恢复。

13.1.1　比较操作

在资源透视图或者 Java 透视图中的文件上右击，选择“比较”→“本地历史记录”就可以很清晰看到各个版本的区别。并可以选择“替换为”→“本地历史记录”选项恢

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

复到以前系统保存的任一版本。版本文件的保存时间，以及文件大小可以在 Eclipse 的“首选项（Preference）”→“工作台（Workbench）”→“本地历史（Local History）”中选项中设置。

图 13.2 是工作空间中的当前版本与本地历史中的另一个版本比较的对话框。



图 13.2　本地历史比较功能

左上角的窗格显示所有可用的版本，按日期排序。右上角的窗格显示编译单元（在本例中为一个类）。在此窗格中，可以选择比较哪些内容。可以比较整个类或单个方法或元素。左下角的窗格显示当前载入的元素，右下角窗格显示右上角中选择的版本。本例比较 main()方法。

替换功能类似于比较功能。但它可以选择另一个版本来替换当前载入的版本。这是一个非常有用的功能。如果希望用文件的先前版本替换当前版本，可以简单地从文件的上下文菜单选择“替换为”→“本地历史记录”。从本地历史替换的窗口如图 13.3 所示。

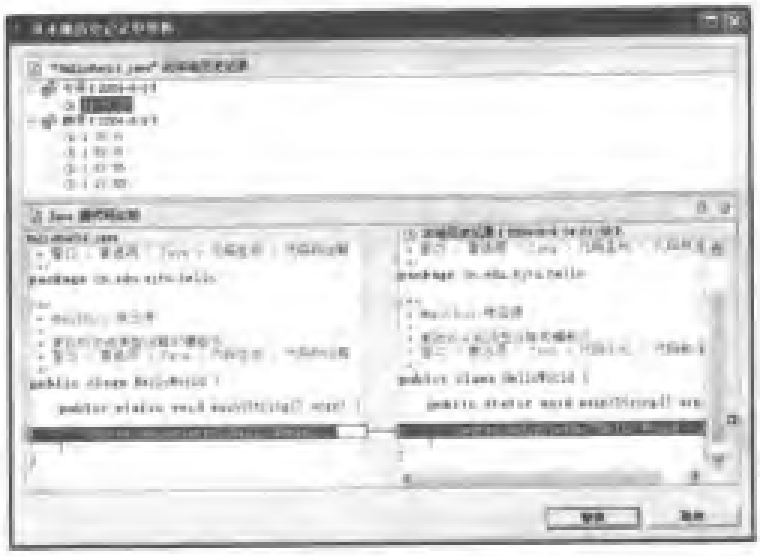


图 13.3　本地历史的替换功能

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

使用 Java 源代码比较窗格右边的箭头可以转到文件的下一个或前一个不同处。单击“替换”将代码从右面窗格移到左面窗格，替换已有的文件。

13.1.2 本地历史记录

在 Eclipse 中还可以比较所有本地编辑的内容。本地历史中的每次编辑都是由保存文件的日期和时间惟一表示的。只有文件才具有本地历史；项目和文件夹不具有本地历史。在 Eclipse 中编辑的文件每次存盘都会留下记录，可以随时与历史记录比较，恢复到某一个时刻的状态。

本地历史的最后一个功能是通过选择所包含的包名来恢复文件中已删除的元素或整个已删除的文件。

选择一个类文件并从它的上下文菜单单击“从本地历史中恢复”后弹出一个对话框，在该对话框中可以看到类的已删除元素并从中选择一些元素恢复到工作台的当前版本中。

可以想像，保留所有修改的本地历史必然要占据硬盘空间。Eclipse 工作台允许配置历史文件保留多长时间、每个文件多少修改保留在历史中以及使用多少硬盘空间。这些设置，如图 13.4 所示，可以通过“窗口”→“首选项”→“工作台”→“本地历史记录”进行查看和设置。

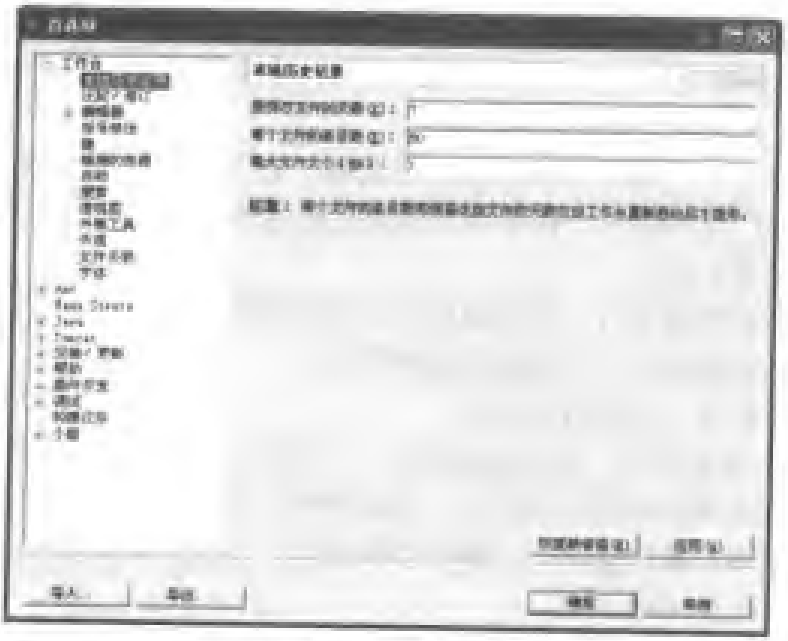


图 13.4 本地历史设置

如图 13.4 所示，对于本地历史记录，可以设置“保存文件的天数”、“每个文件的条目数”和“最大文件大小”项。其中，“保存文件的天数”表示所维护的本地历史记录中的更改的天数，Eclipse 将丢失比此值更旧的历史记录状态，该值的默认值是 7 天。“每个文件的条目数”表示将在本地历史记录中维护的每个文件的历史记录状态的条目数。如果超过了此值，则将丢失更旧的历史记录，以便降低版本维护的存储空间开销，默认

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

的条目数是 50。“最大文件大小”表示在历史记录仓库中个别状态的大小限制。如果文件超过了此大小，则将不会存储它，默认值是 1MB。

本地历史修改保留在元数据中，与项目中的实际文件相分离。文件系统中的项目文件夹中的文件仅包含工作空间中文件的当前版本。

13.2 CVS 概述

Eclipse 内置的本地历史记录只实现了最基本的版本管理功能。然而，目前大多数应用程序是由多人组成的团队开发的。支持团队开发的开发也称小组开发。因此支持小组开发已成为整个开发环境中的重要组成部分。

Eclipse 作为最新的软件开发工具平台已经集成了许多先进的辅助开发功能，如 CVS 版本控制、JUnit 单元测试、Refactoring 代码重构等。

可用于非常庞大的软件开发项目的版本管理系统 CVS（Concurrent Versions System）本身也是一个开放源代码的项目。CVS 是在 UNIX 中的经典的版本控制软件 SCCS 和 RCS 的基础上发展起来的，目前已支持各种不同的 UNIX 操作系统，如 AIX、Solarious、HP-UNIX 等，Linux、FreeBSD 等开放源代码操作系统、Macintosh 以及各种版本的 Windows 操作系统。

CVS 是一个功能强大的版本控制系统。目前已成为开放源代码组织使用的标准版本控制系统，几乎在所有的开放源代码项目中得到应用。由于开放源代码项目的开发者非常多、非常广泛，分布也很散，因此对于版本管理的要求严格。从 CVS 在开放源代码项目中的广泛应用，其功能与性能可见一斑。

虽然 CVS 是一个开放源代码系统，但其除了在开放源代码软件项目如 Apache，GNOME 等中得到广泛使用外，许多非开放源代码的大型软件开发项目亦将其作为重要的软件开发核心管理平台。

CVS 作为源代码管理软件，支持源代码管理软件的两个核心功能：

- 协调对源代码的更改，并能集成这些更改；
- 管理团队所提交工作的历史记录。

CVS 采用 C/S 体系结构，支持远程访问。代码及它的各个版本都存储在一个集中的服务器上。CVS 保存源代码的全部历史、编译环境、运行环境、系统数据、测试系统和软件文档的全部版本。通过建立一个集中的资源库，以及在该资源库上进行版本管理，使得分布式开发者可以协同工作。

在 CVS 资源库中，代码以模块的形式组织起来，每个模块完成一种特定的功能。模块可以是一个程序，可以是程序的一个功能模块，也可以是一个软件包。

当单个开发者开始进行开发时，首先从服务器上获取相关资源的拷贝，在此基础上进行开发。当小组成员完成新的工作时，可随时将新的代码提交给中心资源库。

通过将这些更改提交到资源库，可共享整个团队的工作。类似地，当小组成员希望获得最新可用的工作成果时，就可以根据资源库中的更改，更新自己的本地工作空间。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

这意味着项目资源库会因小组成员提交新的工作成果而经常发生更改。换句话说，资源库应该表示项目的当前状态。任何时候，团队成员都要能够根据资源库更新自己的工作空间，并确定它们是最新的。

在软件开发管理中，对各种各样变化的管理是至关重要的。CVS 将对软件版本的历史记录进行维护，可以将当前工作与先前版本进行比较，如有必要，还可以回复到先前版本。CVS 协调团队的工作，以便只存在惟一的当前项目状态定义，以及集成整个小组的工作。保证整个软件的变化处于控制和管理之下。

13.3 CVS 的基本原理

CVS 系统结构为客户 / 服务器模式。客户端为源代码的工作拷贝，服务器端保存正式提交的源代码及相关的文档、数据，服务器端的这些用于版本管理的信息集合又称为资源库（Repository）。资源库是一个持久仓库，它可以协调由小组开发的资源的多用户访问。

在 CVS 中，所有的文件都用一个同步的版本标记 TAG 来记录，从而可对整个软件的代码进行一体化的版本管理。在 CVS 中，每一个开发者都可以有自己的独立工作空间，可以根据自己的需要将修改随时保存起来，所有的修改都不会丢失。同时，所有的开发者都共享一个主拷贝（Master Copy），任何开发者对文件进行改动之后，其他开发者都可通过更新文件状态了解谁进行了更新、进行了哪些更新。

资源库中的项目可以是两种形式：

- 不可改变——项目版本；
- 可修改——分支中的项目。

CVS 使用分支模型来支持有点相互独立但仍相互依赖的多项工作。分支是协同开发小组共享和集成正在进行的工作的地方。开发小组在分支中共享和集成他们正在执行的工作。将分支看做是小组成员可以随时更新的共享工作区，当小组成员对项目进行更改时就会更新该工作区。此模型允许各个成员从事 CVS 小组项目，在进行更改时与其他成员共享他们的工作以及在项目进展时了解其他成员的工作。因此，个人可以处理小组项目，与小组中的其他人共享他们的工作，在项目的所有阶段访问其他人的工作。在资源库中，称为 HEAD 的一个特殊分支表示工作的主线（HEAD 通常称为干线）。

可以在工作台中更改资源而不影响分支。个人必须显式地将他们更改过的资源提供给分支。在工作台本地进行更改时，开发者的工作是独立于小组的其他成员进行的。当准备让其他小组成员使用开发者的本地资源更改时，需要将工作成果提交至分支。当执行同步时，所有这样的更改被归类为出局（Outgoing）更改。当小组成员完成新的工作时，他们通过将更改提交至分支来共享此工作成果。

同样地，当他们想要获取最新的可用工作成果时，他们将本地工作区更新为分支上的更改。最理想的情况是在提交之前使用其他人所作的所有更改更新本地工作区。这将确保您拥有其他小组成员的最新工作成果。在从分支更新、合并本地“工作台”中的所

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

有冲突更改并在本地测试更改之后，就可以更容易地将“工作台”的更改提交至分支。将更改提交至分支时，更改就会从本地“工作台”复制至分支。结果，当其他开发者稍后从分支更新时，这些更改就被视为入局更改。

因此，当小组成员提交新的工作成果时，分支就在不断地更改并向前发展。分支有效地表示了项目的当前状态。在任何时候，小组成员都可以从分支更新工作区，从而知道工作区是最新的。

CVS 提供在小组中工作所需的两个重要功能部件：

- 小组提交的工作的历史；
- 协调和集成此工作的方法。

维护历史是很重要的，它使得人们能够将当前工作与先前的工作进行比较，恢复为较好的旧工作等。协调工作是很关键的，它使得存在一个当前项目状态定义，包含小组的集成工作。此协调是通过分支模型提供的。

最优模型就是任何小组成员都可以在其中更改他具有访问权的任何资源的模型。由于两个小组成员可以提交至对同一资源的分支更改，所以可能会发生冲突且必须处理它们。此模型称为最优模型，因为假定它发生的冲突很少。

13.4 安装和配置 CVS 资源库

资源库是一个持久化仓库，它可以协调对小组开发资源的多用户访问。Eclipse 工作台中内置了 CVS 支持。可以在以下网址处下载 CVS 服务器：http://www.cvshome.org。

13.4.1 安装 cvsnt

CVS 的安装包括两部分：服务器端的安装和客户端的安装。由于 CVS 一开始主要是基于 UNIX、Linux，因此 Windows 版本的 CVS 服务器 CVSNT 相比其他操作系统下的版本有较大的不同。CVSNT 由相对独立的组织专门开发，该组织的 URL 为：http://www.cvsnt.org。可以从该网站免费下载最新版本，CVSNT 目前最新的稳定版本是 2.0.51d，对应的安装文件是 cvsnt-2.0.51d.exe。

在服务器端用管理员账号安装 CVSNT。CVSNT 的安装很简单，通常每步都可接受默认设置。只是在选择安装方式时，可根据需要选择相应的安装。CVSNT 提供三种安装方式：典型安装“Typical installation”、完全安装“Full installation”和定制安装“Custom installation”，默认值是典型安装，但通常选择完全安装。这种方式将会安装 CVSNT 的服务器端和客户端软件。最后，重新启动计算机，将可完成整个安装工作。

单击“开始”→“程序”→“CVSNT”→“Service control panel”。此时，将出现 CVSNT 服务控制面板，如图 13.5 所示。

安装后，操作系统（Windows2000、Windows2003 等）会自动启动两个服务：CVS Service 和 CVS Lock Service。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

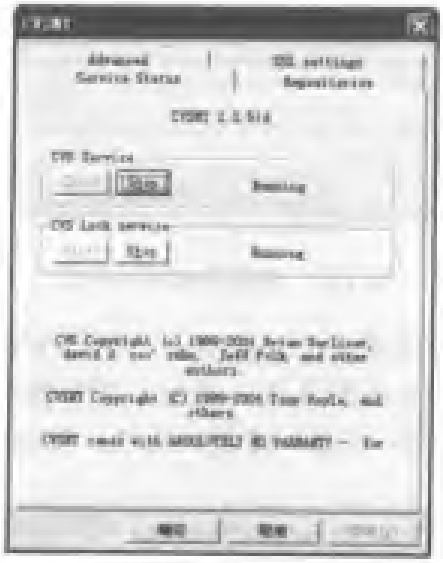


图 13.5 CVSNT 控制面板

可 CVSNT 控制面板对 CVSNT 进行进一步的配置，如添加用户等。需要注意的是，修改 CVSNT 配置时，最好停止服务，等修改结束后，再重新启动。

13.4.2 创建 CVS 资源库

在创建 CVS 资源库之前，首先需要为资源库新建一个文件夹，用于充当存储 CVS 资源库数据的目录。然后，在 CVSNT 控制面板中选择“Repositories”选项卡，就会出现如图 13.6 所示的对话框窗口。

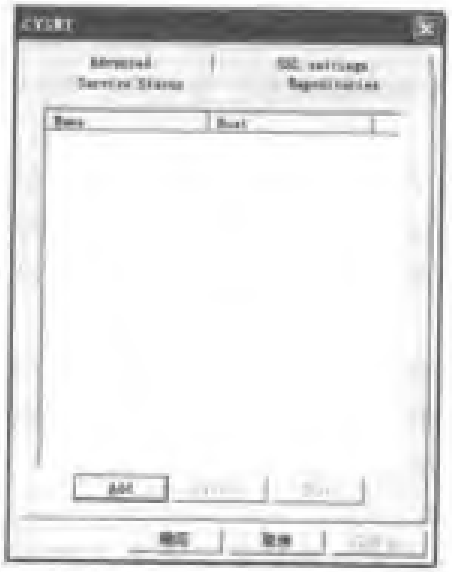


图 13.6 Repositories 选项卡

在 Repositories 选项卡窗口中，单击“Add”按钮，进入资源库的编辑界面，如图 13.7 所示。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理



图 13.7 编辑资源库的位置

在资源库编辑窗口中，可直接在“Location”字段中输入资源的目录，也可单击该字段右侧的“…”按钮，选择合适的目录。例如，在本例中，我们选择为资源库所创建的“E:\cvsroot”目录，然后单击“OK”按钮。此时，将会弹出如图 13.8 所示的窗口。



图 13.8 CVS 仓库目录的初始化

单击“是”按钮，相应的条目将出现在 Repositories 选项卡下的窗口中。此时，将完成 CVS 资源库的设置。单击“确定”按钮，退出 CVS 服务控制面板。

13.5 Eclipse 中实施 CVS 系统

Eclipse 可以支持多个软件管理和版本控制产品如 Rational、CA、Merant 等公司的产品。但 Eclipse 中使用最广泛的版本控制系统是 CVS。可以在以下网址处下载 CVS 服务器：http://www.cvshome.org。

Eclipse 默认为版本管理工具 CVS 提供了接口，可以非常方便地连接到 CVS 服务器上。通过 CVS 版本管理，Eclipse 为团队开发提供良好的环境。在 CVS 编程环境中，小组成员在他们自己的工作台中独立完成所有工作，他们可以通过 CVS 资源库共享他们的工作。

除非将所创建项目等资源提交至资源库，否则小组中的其他人将无法一起使用这些资源。要使其他小组成员也能够共享资源，需要执行两项任务：

① 指定小组的 CVS 资源库的位置；

② 将工作提交至该资源库。

在 Eclipse 中使用 CVS 需要注意的是一些术语和功能的变化，CVS 中的“分支（Branch）”在 Eclipse 中被称为“流（Stream）”，取消了 CVS 中“检出（check out）”、“导入（import）”和“提交（commit）”等功能，统统用鼠标右键菜单中的“小组（Team）”→“流同步（Synchronized with Stream）”来替代。这些功能都通过图形界面完成，在每

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

次操作中都会有当前文件和以前各个版本的比较窗口，操作非常直观，易于掌握。

在 CVS 小组编程环境中，小组成员在他们自己的 Eclipse 工作台中执行所有工作，与其他人分开。最后，他们将共享他们的工作。他们通过“CVS 资源库”来完成此操作。工作台中内置了 CVS 支持。CVS 资源库是一个持久数据存储，它可以协调对项目及其内容的多用户访问。可通过局域网或广域网以实现资源库与“工作台”客户机之间的通信。CVS 远程访问时，必须进行验证并登录。登录 CVS 后，才可以进行各种操作。目前“工作台”支持两种用于 CVS 的内部认证协议：pserver 和 ssh。常用的方式是 pserver。CVS 还可以使用外部工具提供的认证协议。

通常，资源不会孤立存在：它们通常包含其他资源的隐式或显式相关性。例如，Web 页具有指向其他 Web 页的链接。将资源提交至分支时，这些相关性可能会受到影响。由于分支表示当前项目状态，从而确保相关性的完整性是很重要的：在任何时候，小组成员都可以将分支内容用做新工作的基础。因此，理想的工作流就是保留分支完整性的工作流。理想工作流流程如下。

① 开始：开始工作之前，用当前分支状态更新工作区中的资源。如果你确定没有你关心的本地工作，则最快的获取方法是从分支（或 HEAD）中选择您感兴趣的项目并选择作为项目检出（或者如果项目在本地已经存在，则选择替换为资源库中的最新内容）。这将用分支中的内容覆盖本地资源。

② 进行更改：在工作台本地工作，创建新的资源，修改现有资源以及完成时在本地保存资源。

③ 同步：当准备好提交工作成果时，与资源库同步。

a. 更新：检查入局更改并将它们添加至本地工作台。这允许用户确定是否存在可能影响将提交内容的完整性的更改，解决冲突。重新测试并运行完整性检查器（例如，检查断开的超文本链接及确保代码编译等）。

b. 提交：如确信更改已与最新的分支内容很好地集成在一起，就将更改提交至分支。为谨慎起见，可重复先前的步骤（如果有新的入局更改）。

在某些情况下，可能入局更改对用户毫无影响，从而选择提交而不进行更新。然而，小组成员通常应尽量遵循类似上述的流程来确保不会意外危及分支完整性。

下面我们将详细讨论 Eclipse 中实施 CVS 的具体问题，例如：如何使用项目、将更改提交至资源库以便小组中的其他成员使用；当我们与其他用户一起使用项目时，我们将了解如何提交其他用户可能同时在使用的资源，以及如何使用其他人在项目中所作的更改更新工作区等。

13.5.1　向资源库提交项目

除非将项目提交至资源库，否则小组中的其他人将无法与我们一起使用该项目。要使其他小组成员也能够使用此项目，需要执行三项任务：

① 指定小组的 CVS 资源库的位置；

② 项目同步；

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

③ 将工作提交至该资源库。

1. 指定小组的 CVS 资源库的位置

必须先指定可用资源库，才能与其他用户共享我们的项目。在 Eclipse 中指定 CVS 资源库的位置的步骤如下。

① 依次选择菜单“窗口”→“打开透视图”→“其他”→“CVS 资源库浏览”，打开“CVS 资源库浏览”透视图。如图 13.9 所示，左上方的视图显示当前使用的所有 CVS 资源库。

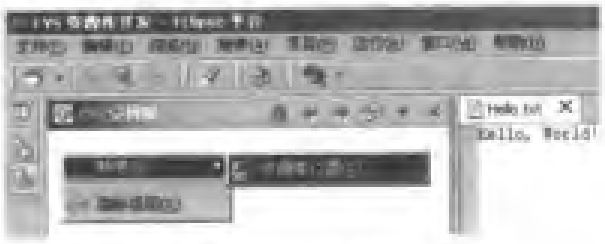


图 13.9 指定资源库位置

② 如 CVS 资源库视图是空的，这意味着还需要指定资源库。在“CVS 资源库”视图的上下文菜单中，选择“新建”→“资源库位置…”，将打开“添加 CVS 资源库窗口”，如图 13.10 所示。



图 13.10 添加 CVS 资源库

③ 在 CVS 资源库位置向导中，填充资源库的位置以及登录信息。在主机字段中，输入主机的地址；在资源库路径字段中，输入在主机地址处的资源库的路径；在用户字段中，输入想要用来进行连接的用户名；在密码字段中，输入密码；在连接类型字段中，选择资源库的 CVS 连接类型（默认类型为 pserver）；将使用默认端口保留为启用状态。默认情况下会选择完成时验证位置。如选择完成时验证位置，向导将连接至资源库以验证该信息。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

④ 完成上述步骤后单击“完成”按钮，资源库视图则显示新的资源库位置，如图 13.11 所示。



图 13.11 CVS 资源库

“CVS 资源库”视图显示已添加至“工作台”的 CVS 资源库位置。展开某个位置会显示该资源库中的主干（HEAD）、项目版本和分支。可进一步展开项目版本和分支以显示它们当中所包含的文件夹和文件。

“CVS 资源库”视图的弹出菜单还允许用户指定新的资源库位置。使用“CVS 资源库”视图将资源从资源库检出至工作台，配置视图显示的分支和版本，查看资源历史及比较资源版本。与工作台中的其他视图一样，CVS 资源库视图有它自己的工具栏。提供工具栏按钮是为了刷新视图和浏览。工具栏还包含下拉菜单，允许用户切换视图以显示包含在资源库中的文件夹或在 CVSROOT / 模块文件中定义的模块。

资源库位置不是实际的现场连接。而是对资源库所在位置的描述。稍后，当我们将工作成果提交至资源库或使用其他人完成的工作成果进行更新时，工作台将根据此位置信息创建连接。

2. 项目同步

一旦配置好 CVS 资源库，就可以将项目输入资源库。在创建了项目并且指定了资源库位置后，在该位置和 HEAD 分支共享项目。需要通过如下步骤将项目同步。

① 在“导航器”视图中选择项目。从项目的上下文菜单中，选择“小组→共享项目”。

② 如果安装了多个资源库提供程序，则选择 CVS 并单击“下一步”按钮。在共享向导页面中，选择先前创建的位置。

③ 完成上述步骤后单击“完成”按钮，则同步视图自动打开。

同步视图提供了一个位置，可在其中查看用户所作的更改和其他人做出的并已提交至资源库的更改。除提供复合本地、远程资源树之外，它还允许用户将更改提交资源库，并获取其他人做出的更改。默认情况下，如果与资源库同步，而资源库中没有可供采用的更改，则会自动处于出局方式以允许快速提交更改。

3. 将工作提交至该资源库

可以将已修改的工作台资源提交给资源库，以便其他协同开发成员可以查看所作的工作。只有在该分支上工作的其他成员才能看见在该分支上提交的那些更改。工作提交可以按两种不同的级别：文件级和项目级。

① 如果提交个别文件，在“结构化比较”列表中选择所需提交的文件，通过在“结构化比较”列表中选择资源，底部区域将显示我们在工作台中所具有的资源与当前在资源库的资源之间的差别。从弹出菜单中选择“提交”，当进行添加并提供提交注释时单击“确

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

定”按钮。

② 除了提交文件，还可以选择该项目并提交该项目中的所有更改。从项目的上下文菜单中选择提交。

可以从两个位置发出提交命令将工作提交至资源库，供其他成员共享使用：小组→提交菜单，或者同步视图。

（1）使用“小组→提交”提交更改

① 在“导航器”视图中，选择想要提交的资源。

② 右键单击该资源，并从弹出菜单中选择小组→提交。

③ 在“提交注释”对话框中，提供对更改的注释（例如，修正拼写错误）。

如果更改采用以上方式提交的任何文件时都出现了冲突，则操作将失败。如果发生这种情况，则必须执行更新，或者使用“同步”视图来解决冲突。为了确保在提交更多更改之前具有资源库的最新状态，在以任何方式提交之前都必须进行更新。

（2）在同步视图中提交更改

① 在“导航器”视图中，选择想要提交的资源。

② 单击右键打开弹出菜单，并选择小组→与资源库同步，将打开“同步”视图。

③ 在“同步”视图的工具栏上，选择出局方式按钮以显示可能具有的已修改的任何“工作台”资源（出局更改）。

④ 如果存在冲突（红色箭头），则应解决这些冲突。使用文本比较区域来将资源与冲突进行合并。可以将更改从文件的资源库修订版拷贝到文件的“工作台”修订版，并保存已合并的“工作台”资源。一旦解决了“结构比较”区域中的所有冲突，就可以提交了。

⑤ 在“结构比较”窗格中，右键单击想要提交的层次结构的顶部，并从弹出菜单中选择提交。

⑥ 在“提交注释”对话框中，提供对更改的注释（例如，修正拼写错误）。

可以通过执行覆盖并提交来提交具有冲突的文件。这将把资源的“工作台”副本提交到资源库中，从而除去任何入局更改。

以上描述的覆盖并提交命令的行为只适用于同步视图的出局方式。在视图的入局/出局方式中，出局更改和冲突的行为是相同的，但是该命令将把入局更改回复为本地工作台的内容。

（3）三方比较

三方比较显示资源的三个不同版本之间的差别。当合并资源或同步期间存在冲突时，此功能部件最为有用。如果两个开发者从同一分支向他们的工作台添加版本，然后每个开发者都修改它，其中一个开发者试图在另一个开发者已提交资源之后提交它，则会发生冲突。

当发生这种情况时，可查看三个资源版本之间的差别：工作台中的资源、在分支中提交的资源的版本及公共祖先（两个冲突版本都基于该公共祖先）。（如果不能确定公共祖先，例如，由于两个不同的开发者创建并提交名称和路径相同的资源，比较就变成了两方比较。）同步视图允许查看两个或三个文件之间的差别。如果公共祖先可用，则同步

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com

教程由站长百科收集整理

视图会执行三方比较。可能不能确定两个冲突资源版本的公共祖先（例如，两个不同的开发者创建并提交名称和路径相同的资源）。在这种情况下，比较就成了一般的两方比较。在三方比较中，工作台显示：

- 与公共祖先相比，第一个子代更改了什么内容；
- 与公共祖先相比，第二个子代更改了什么内容。

在三方比较中，公共祖先显示在文本比较窗格的顶部窗格中。突出显示的差别就是“工作台”资源中与公共祖先比较起来已经更改的内容以及分支资源中与公共祖先比较起来已经更改的内容。在所有三个文件中不相同的部分会突出显示以示存在差别。冲突以红色显示，入局更改以蓝色显示，而出局更改以灰色显示。

13.5.2 标记资源版本

CVS 中的资源是通过使用版本标号标记来设置版本的。设置资源版本以在特定位置及时捕获资源当前状态的快照。当设置资源版本时，就意味着可在资源库中检索该资源的不可修改副本。设置项目版本将保存项目中所有资源版本的序列。可设置除项目之外的资源（即文件夹和文件）的版本。然而，更常见的做法是一起设置整个项目的版本，因为项目中包含的资源的相互依赖程度很高。可从工作区或从 CVS 资源库视图中的分支（包括 HEAD）设置项目版本。这两种方法之间的差别在于决定哪些子资源版本应是项目版本的一部分。

当从工作台将项目标记为版本时，“工作台”中文件的基本修订版就会被标记为属于该版本。由于您确切地知道该版本中将有哪些文件修订，因此这是设置项目版本的最好方法。当具有出局更改或未提交更改时允许使用此方法。只会忽略未提交更改，而对于具有出局更改的资源，仍然可以让它们的基本版本作为版本的一部分。如果需要在开始更改资源的位置分割项目并将资源提交至另一个分支，则对具有未提交更改或出局更改的项目设置版本是很方便的。

1. 在工作区创建项目版本

当从“CVS 资源库”视图中的分支将项目标记为版本时，就会即时对分支中当时最新的资源版本的内容设置版本。如果不知道在分支中提交了哪些内容，则不能从分支中设置项目的版本。因此，通常选择从“工作台”设置版本。

想要设置版本的项目首先必须位于工作区中。通常选择在设置版本之前使资源与 CVS 资源库同步。这将确保不会从该版本意外地排除未完成的入局或出局更改。位于工作区的项目可如下设置版本。

① 在“导航器”视图中，选择想要设置版本的项目。

② 从项目的上下文菜单中，选择小组→标记为版本。

③ 输入想要创建的版本的名称。

④ 单击“确定”按钮以关闭标记资源对话框并创建版本。

可以通过单击“标记资源”对话框中的详细信息按钮来浏览项目的现有版本。单击

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

“从资源库刷新”将显示“现有版本”列表。在命名新版本之前查看现有版本时非常有用。

2. 设置资源库中项目的版本

可以设置资源库中项目的版本而不必将项目添加到工作台。

① 切换至“CVS 资源库管理”透视图，或者将“CVS 资源库”视图添加至当前透视图。

② 在“CVS 资源库”视图中，展开资源库位置，并查找想要为其设置版本的项目。

③ 右键单击该项目，并选择标记为版本。

④ 输入想要创建的版本的名称。

⑤ 单击“确定”按钮，关闭对话框并创建版本。

如果想要应用的版本标记在要设置版本的某些资源上已经存在，则可以执行使用现有标记。此操作将把现有标记移至所选的资源，并且可用来移动版本和分支标记。如果想要应用的标记未显示在列表中，则可以单击从资源库刷新。如果这样也找不到标记，则可以单击配置标记，这将打开一个对话框，它允许用户搜索资源库中特定文件上的标记。

13.5.3 从 CVS 资源库检出项目和模块

任何文件夹，包括非根文件夹，都可以从 CVS 资源库检出。如下可将项目从 CVS 资源库检出至工作台。

① 切换至“CVS 资源库浏览”透视图或将“CVS 资源库”视图添加至当前透视图。

② 在“CVS 资源库”视图中，展开资源库位置。

③ 展开 HEAD 并选择想要作为项目添加至“工作台”的文件夹。如果要查找特定版本的文件夹：

a. 展开版本类别并查找想要添加至“工作台”的文件夹名。

b. 展开文件夹以查看它的版本。

如果要查找分支中的最新文件夹：

a. 展开分支类别。

b. 展开存放要添加至“工作台”的文件夹的分支。

如果没有看到期望的文件夹，则通过从视图标题栏中的下拉菜单选择显示文件夹来确保 CVS 资源库视图处于显示文件夹方式。

④ 从选择的文件夹的弹出菜单，选择下列项之一：

a. 作为项目检出，将每个选择的文件夹作为本地工作区中与资源库中的文件夹具有相同名称的项目检出。

b. 将选择的文件夹检出至本地工作区内定制配置的项目。仅对单个文件夹才启用此选项。

⑤ 如果选择了检出至本地工作区内定制配置的项目，则根据资源库中的文件夹是否包含 .project 文件，将显示两个可能的对话框之一。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

a. 如果有.project 文件，则对话框将接受定制项目名称和位置。

b. 否则，对话框将为新项目向导，它允许项目（例如 Java 项目）的完全定制。

可以在 CVS 资源库的 CVSROOT/modules 文件中定义模块。如下可将此类模块从 CVS 资源库检出至工作台。

① 切换至“CVS 资源库浏览”透视图或将 CVS 资源库视图添加至当前透视图。

② 通过选择视图的标题栏中的下拉菜单中的显示模块来将“CVS 资源库”视图切换至模块方式。

③ 在 CVS 资源库视图中，展开资源库位置。

④ 展开 HEAD 并选择想要添加至“工作台”的模块。

⑤ 从弹出菜单选择检出模块。根据模块在资源库中的定义，将把模块检出至本地工作区中的一个或多个项目中。

13.5.4 更新

当用户对工作台中的项目进行工作时，小组中的其他成员可能正在提交对资源库中该项目副本的更改。要获取这些更改，可以更新工作台以便与分支的状态相匹配。更改将特定于工作台项目被配置为可共享的分支。开发者可以控制何时选择进行更新。可以从两个位置发出更新命令：小组→更新菜单，或者同步视图。为了理解这两个命令之间的差别，了解三种不同类型的入局更改是很重要的：

- 非冲突更改。当已经以远程方式更改了一个文件，但是尚未在本地修改该文件时，就会发生这种更改。
- 可自动合并的冲突更改。当已经以远程和本地方式（即具有未提交的本地更改）更改了 ASCII 文件，但是更改位于不同行上时，就会发生这种更改。
- 不可自动合并的冲突更改。当已经以远程和本地方式更改了 ASCII 文件中的一个或多个相同行或者二进制文件时（二进制文件从来就不是可自动合并的），就会发生这种更改。

当选择“小组”→“更新”时，本地资源的内容将更新为所有以上三种类型的入局更改。对于非冲突和可自动合并的冲突，不需要执行附加操作（对于可自动合并的冲突，仅当用户不想进行自动合并时，才会将已更改的本地资源移至前缀为“.#”的文件中）。然而，对于不可自动合并的冲突，会使用特殊的 CVS 特定标记文本将冲突合并到本地资源中（对于 ASCII 文件），或者将已更改的本地资源移至前缀为“.#”的文件（对于二进制文件）。这与 CVS 命令行的行为相匹配，但是在与 Eclipse 自动构建机制组合时可能会有问题。另外，在更新任何本地资源之前，通常期望知道存在哪些入局更改。这些问题由“同步”视图解决。

以入局方式打开“同步”视图的步骤如下。

① 在“导航器”视图中，选择想要更新的资源。

② 从所选资源的弹出菜单中，选择“小组”→“与资源库同步”，将打开“同步”视图。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

③ 在“同步”视图的工具栏上，单击“入局方式”按钮，过滤掉可能已修改的任何工作台资源（出局更改）。

在入局方式中，将看到自从上次更新以来已提交给分支的更改。视图将指示每种入局更改的类型（非冲突、可自动合并冲突或不可自动合并冲突）。有两个更新命令（可从视图中的任何资源的上下文菜单中获得）用来处理不同类型的冲突：从资源库更新和覆盖并更新。当在“同步”视图中选择从资源库更新命令时，将只处理非冲突更改，使具有可自动合并或不可自动合并冲突的任何文件保留在视图中（将从视图中除去已成功处理的任何文件）。覆盖并更新命令对两种类型的冲突执行操作。在选择此命令之后，在尝试执行合并之前将提示并询问您是想要自动合并内容还是用资源库文件覆盖它们。如果选择自动合并，则将只处理可自动合并的冲突，并且会自动将入局更改与本地更改合并。否则，将处理所有冲突，并且将把本地资源替换为远程内容。通常不期望发生此替换行为。下面描述了一种替代方法，更新非冲突和可自动合并的文件，步骤如下。

① 在“同步”视图顶部的“结构比较”窗格中包含具有入局更改的资源的层次结构。

② 选择非冲突文件，并从弹出菜单中选择从资源库更新。这将更新所选择的资源并从视图中除去它们。

③ 选择可自动合并的冲突，并从弹出菜单中选择覆盖并更新。选择只更新可自动合并的资源，并在提示时单击“确定”按钮。这将更新所选择的资源，并从视图中除去它们。

如果本地工作台包含不可与来自分支的入局更改自动合并的任何出局更改，则不执行覆盖并更新，开发者可以将差别手工合并到工作台中，步骤如下。

① 在“结构比较”窗格中，如果资源列表中存在冲突（用红色箭头表示），则选择它。

② 在“同步”视图的“文本比较”区域中，在左边表示的是本地工作台数据，在右边表示的是资源库分支数据。检查两边数据的差别。

③ 使用文本比较区域来合并任何更改。可以将更改从文件的资源库修订版复制到文件的“工作台”副本，并保存合并的“工作台”文件（使用左窗格中的弹出菜单）。

④ 一旦完成了将远程更改合并到本地文件中，就从弹出菜单中选择标记为已合并。这将把本地文件标记为已更新，并且允许开发者提交更改。

当进行更新时，不会更改资源库内容。当接受入局更改时，这些更改将应用于工作台。仅当用户提交出局更改时才会更改资源库。在“结构比较”窗格中，选择一组入局更改的祖先时将对所有适当的子代执行操作。例如，如果选择最顶层的文件夹并选择从资源库更新，则将处理所有非冲突的入局更改，并且不处理所有其他入局更改。

以上描述的覆盖并更新命令的行为只适用于“同步”视图的入局方式。在视图的入局/出局方式中，入局更改和冲突的行为是相同的，但是该命令将把出局更改恢复为资源库的内容。

13.5.5 补丁程序的创建和应用

当开发者想要通过资源库添加共享的项目，但又不具有对资源库的写存取权时，可

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

以使用补丁程序技术。补丁程序允许开发者共享工作而不必将它存储在资源库中。

开发者可以创建补丁程序并根据项目定义的进程，用电子邮件将其发送到具有写存取权限的开发者或将其附加到项目使用的故障报告系统中的故障中。具有写存取权限的开发者就可以将补丁程序应用到项目并提交更改。

在 CVS 项目中创建补丁程序的步骤如下。

① 选择包含要包括在补丁程序中的修改资源。必须将补丁程序应用到生成它的同一资源中。还应将补丁程序应用到生成它的同一文件修订上，所以应确保将补丁程序应用到同一资源队列，完成此操作的最简单方法是在某版本顶层创建补丁程序。

② 从弹出菜单中，选择“小组”→“创建补丁程序…”，这将打开“创建补丁程序”向导。

③ 选择要保存补丁程序的位置：

a. 保存至剪贴板——这将把补丁程序放置在剪贴板上，以便可以将其粘贴到文本编辑器，如电子邮件程序中。

b. 保存至文件系统——这将把补丁程序放置到本地文件系统中指定的文件中。

c. 在工作区中保存——这将把补丁程序放置到某个现有工作台项目内指定的文件中。

对于小补丁程序，在大多数情况下，使用本地文件系统是最好的选择。单击“下一步”按钮，配置生成补丁程序的选项。

④ 选择如何配置补丁程序：

a. 递归至子文件夹中——如果禁用，则仅将选择的直接子代包括在补丁程序中。否则，将包括所有后代。

b. 将新文件包括在补丁程序中——如果禁用，则仅包括处于 CVS 版本控制之下的文件。否则，将还包括新创建但尚未添加或已忽略的文件。

c. Diff 输出格式——允许选择几个 comon diff 输出格式。Unified 是由许多补丁应用程序工具（包括 Eclipse）使用的格式。

⑤ 单击“完成”按钮。

⑥ 对正在修补的项目以适当的方式发送补丁程序。

应用补丁程序的步骤如下。

① 选择生成补丁程序所在的资源。此资源应包含与生成补丁程序所在的队列相同的文件修订。

② 在上下文弹出菜单中，选择“比较”→“补丁程序…”，这将打开“资源修补器”向导。

③ 指示将查找补丁程序的位置：

a. 文件——补丁程序在本地文件系统上的文件中。输入文件的全路径或使用“浏览”按钮查找文件。

b. 剪贴板——补丁程序在剪贴板上。使用基于文件的补丁程序更安全。如果使用剪贴板并且补丁程序是在其他平台上生成的，如 Linux 与 Windows 等，则可能不会正确地处理行结尾。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

单击“下一步”按钮，查看应用补丁程序的效果。

④ 此页面的顶窗格显示补丁程序是否可以成功地应用到工作区中的文件。如果在树中选择了叶项，则底部窗格将以并列方式显示补丁程序文件的这一部分，在补丁程序术语中称之为“块”。

a. 选择的项指示补丁程序（或块）可以成功应用到工作区资源。可以通过取消选择补丁程序或个别的块来排除它们。

b. 红色叹号指示补丁程序或块有问题。如果补丁程序的格式不是良好的，或生成补丁程序所在的一个或多个文件与补丁程序要应用至的修订不匹配，则将发生这种情况。

c. 要成功地应用全路径，开发者将必须解决问题（红色叹号表示）并通过处理此向导页面上的选项来应用选择的项。

⑤ 如果没有问题，单击“完成”按钮应用补丁程序。

工作区就会将包含补丁程序修改的每个文件的出局更改。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 14 章　Eclipse 中的资源构建和构建工具 Ant

大型系统的构建工作往往是一个极其繁冗和枯燥的过程。通常需要借助构建工具构建和部署应用。

14.1　Ant 简介

Ant 是一个基于 Java 的跨平台的构建工具。类似于 UNIX、Linux 下的 Make 工具，以批处理的形式执行各种任务，如文件拷贝、编译、打包及自定义任务。

14.1.1　Ant 概述

Jakarta 的 Ant 是很好的构建工具。Ant 项目和 Apache 软件基金会的其他项目一样也是一个开放源代码项目。

Ant 使用 XML 来编写构建脚本程序。build 文件组织为需要执行的目标。每一个目标都可能依赖于其他的目标。Ant 将根据所请求执行的目标，以及一组依赖目标，来执行任何需要的目标。每一个目标都可能包含任意数量的 Ant 任务，而由 Ant 任务实际执行目标的工作。Ant 有很多内置的任务，可以完成诸如编译 Java 代码、生成文档、操纵文件和目录等任务。同时，第三方又提供了很多附加的任务。

Ant 易于理解、使用和扩展，已经成了事实上的 Java 构建标准。现在，几乎所有的开源 Java 项目都在使用 Ant，许多公司的开发项目也在使用 Ant。有关更多详细信息，参见 http://ant.apache.org/index.html。

14.1.2　Ant 的获取和安装

因为 Ant 是开放源代码软件，所以任何人都可以自由地得到 Ant 软件，包括二进制版本和源代码版本。

Ant 的二进制稳定版可从 http://ant.apache.org 下载，目前的最新版本是 1.6.1。稳定版本的源码的下载地址为：

```
http://jakarta.apache.org/builds/ant/release/v1.1/src
```

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

最新版本的源码的下载地址为：

```
http://jakarta.apache.org/from-cvs/jakarta-ant/
```

如果下载的是源代码版本，首先需要将其编译为二进制软件。如果是二进制版本，则其安装非常简单，直接将下载文件解压即可。Ant 解压后的目录结构包括三个目录：bin、docs 和 lib。只有 bin 和 lib 目录是运行 Ant 所需的。Ant 软件所位于的目录称为 Ant 的安装目录，或称做 ANT_HOME。

安装工作还包括 Ant 环境变量的设置，需要将 ANT_HOME 和 ANT_HOME\bin 设置为环境变量。

可在下载 Ant 的同时，下载可选包。目前包含可选任务的 JAR 文件名为 jakarta-ant-1.6.1-optional.jar。将这个 JAR 文件放到 Ant 安装目录的 lib 目录下，从而使得 Ant 可以支持一些可选的任务。

14.1.3 Ant 的配置

为了让 Ant 与 Tomact 管理软件协同工作，需要将 catalina-ant.jar 从 Tomcat 的 server/lib 目录拷贝到 Ant 的 lib 目录下。将数据库 JDBC 驱动程序复制到 Tomcat 的 common/lib 目录下使 Tomcat 和其上的 Web 应用程序可以访问数据库。最后，必须建立 Tomcat 的 admin 和 manager 角色，以及其用户名和密码。编辑 Tomcat 的 conf 目录下的 tomcat-user.xml 文件如下：

```
<?xml version='1.0'?>
  <tomcat-users>
     <role rolename="admin"/>
     <role rolename="manager"/>
     <user username="tomcatusername" password="tomcatpassword"
           roles="admin,manager"/>
  </tomcat-users>
```

14.2 构建文件

当开始一个新项目时，首先应编写 Ant 构建文件。换句话说，每个构建文件含有一个项目。文件定义构建的过程，并为团队中的每个开发人员所使用。

Ant 构建文件 build.xml 可以放在任何位置，但是放在项目顶层目录中可以保持项目的简洁。这是最普遍的规范，使开发者能够在根目录中找到它。同时，也能够容易了解项目中不同目录之间的逻辑关系。以下是一个典型的项目层次：

```
[根目录]
  | build.xml
  +--src
  +--lib (包含第三方 JAR 包)
  +--build (由 build 任务生成)
  +--dist (由 build 任务生成)
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

如图 14.1 所示，每个项目可以定义一个或多个目标，以及任意数目的属性目标有可能会依赖于其他目标。当执行这类目标时，需要首先执行它们所依赖的目标。

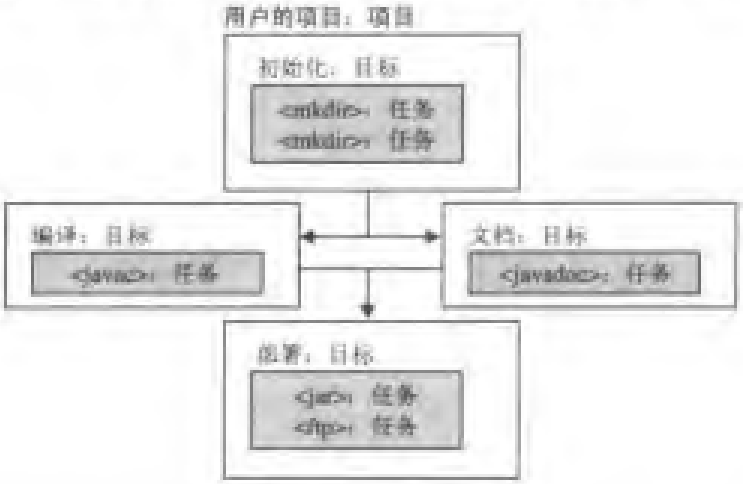


图 14.1 项目的组成

在目标中可定义多个任务，目标定义了所要执行的任务序列，Ant 在构建目标时必须调用所定义的任务。任务定义了 Ant 实际执行的命令。Ant 中的任务可分为三类：

- 核心任务

核心任务是 Ant 自带的任务。

- 可选任务

可选任务是来自第三方的任务，因此需要一个附加的 JAR 文件。

- 用户定义的任务

用户定义的任务是用户自己所开发的任务。

下面是一个构建文件的示例：

```
<?xml version="1.0" ?>
<project name="MYPROJECT" default="compile" >

    <target name="init">
        <mkdir dir="build/classes" />
        <mkdir dir="bin" />
    </target>

    <target name="compile" depends="init">
        <javac srcdir="src" destdir="build/classes" />
    </target>

    <target name="archive" depends="compile">
        <jar destfile="bin/myproject.jar"
            basedir="build/classes" />
    </target>

<target name="execute" depends="compile">
        <java
            classname="FirstClass"
            classpath="build/classes">
            <arg value="first argument passed"/>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
            <arg value="second argument passed"/>
        </java>
    </target>

    <target name="clean" depends="init">
        <delete dir="build" />
        <delete dir="bin" />
    </target>

</project>
```

构建文件通过高层的标记进行定义，从而使得构建文件易于理解。下面将解释构建文件中所用的主要标记及其具体含义和用法。

14.2.1 <project>

在项目标记<project>中可定义多个属性，例如：

```
<project name="MYPROJECT" default="compile" basedir=". "
         description="This is a demo project">
```

各个属性的含义分别是：

- name，项目名
- default，默认的运行目标
- basedir，项目的基准目录
- description，项目的描述

每一个构建文件都对应于一个项目。然而，大型项目经常包含大量的子项目。每一个子项目都可以有自己的构建文件。

在一个构建文件中可以定义多个目标，每个目标都可用于表示构建的不同输出，并且每一个目标都可能依赖其他的目标。

14.2.2 <target>

目标中可能包含多个任务，任务是目标的实际工作。目标也可定义多个属性，例如：

```
<target name="compile" depends="init" description="Compile everything">
```

在目标中可定义的属性包括：

- name，目标名
- depends，依赖的目标
- if，仅当属性设置时才执行
- unless，仅当属性没有设置时才执行
- description，项目的描述

一个 target 可以依赖于其他的 target。例如，可能会有一个目标用于编译程序，一个

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

目标用于生成可执行文件。在生成可执行文件之前必须先编译通过，所以生成可执行文件的目标依赖于编译目标。Ant 会处理这种依赖关系。

Ant 的 depends 属性指定了目标应该被执行的顺序。Ant 会依照 depends 属性中目标出现的顺序（从左到右）依次执行每个目标。在执行目标前，首先需要执行它所依赖的目标。然而，如果被依赖的目标无法运行，这种 depends 对于指定了依赖关系的目标就没有影响。例如：

```
<target name="A"/>
<target name="B" depends="A"/>
<target name="C" depends="B"/>
<target name="D" depends="C,B,A"/>
```

其执行顺序是 A→B→C→D，而不是 C→B→A→D。

一个 target 只能被执行一次，即使有多个 target 依赖于它。如果没有 if 或 unless 属性，target 总会被执行。可选的 description 属性可用来提供关于 target 的一行描述，这些描述可由-projecthelp 命令行选项输出。

14.2.3 <mkdir>

该标签用于创建一个目录，如：

```
<mkdir dir="dist/bin"/>
```

其中，dir 属性指定了所创建的目录名。

14.2.4 <jar>

该标记用于生成一个 JAR 文件，例如：

```
<jar destfile="dest/working.jar">
    <include name="**/*.class"/>
    <exclude name="**/*hello*"/>
</jar>
```

该标记中可定义的属性有：

- destfile，JAR 文件名
- basedir，被归档的文件目录
- includes，被归档的文件模式
- excludes，被排除的文件模式

14.2.5 <javac>

该标记用于编译一个或一组 Java 文件，例如：

```
<javac srcdir="src" destdir="bin"/>
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

该标记的属性包括：

- srcdir，源程序树的基准目录
- destdir，输出目录
- includes，被编译的文件的模式
- excludes，被排除的文件的模式
- classpath，所使用的类路径
- debug，包含调试信息
- optimize，使用优化
- verbose，提供详细的输出
- fileonerror，当碰到错误时自动停止

14.2.6 <java>

该标记将执行 Java 程序（.class 文件），例如：

```
<java classname="mydemo"/>
```

该标记的属性包括：

- classname，将运行的类名
- jar，包含该类的 JAR 文件名
- classpath，所用到的类路径
- fork，在一个新的虚拟机中运行该类
- failonerror，当出现错误时自动停止
- output，输出文件
- append，追加或者覆盖默认文件

14.2.7 <delete>

该标记用于删除一个文件或一组文件，例如：

```
<delete file="myfile.java" />
```

该标记的属性包括：

- file，所要删除的文件
- dir，所要删除的目录
- includeEmptyDirs，指定是否删除空目录，默认值是删除
- failonerror，指定当碰到错误时是否自动停止，默认值是自动停止
- verbose，指定是否列出所删除的文件，默认值时不列出

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

14.2.8 <echo>

该标记用于将消息写到 System.out，有两种表示方式，例如：

```
<echo message="Time to go" />
<echo> Time to go </echo>
```

该标记的属性包括：

- message，输出的消息文本
- file，输出文件
- append，追加到文件后，而不是覆盖文件

14.2.9 <property>

属性定义包含名字－值对，其中 name 表示属性名，value 表示属性的值，例如：

```
<property name="report_location"  value="C:/reportdir "/>
```

Ant 预定义的属性有：java.class.pat，os.name，os.version，user.name，user.home，ant.version，ant.file，ant.project.name 等。

对属性可进行各种操作，例如：

```
<delete dir="${report_location}"/>
```

14.2.10 <buildnumber>

该标记从文件中查看构建数，并设置 build.number 属性，例如：

```
<buildnumber file="buildnumberfile.txt"/>
```

其中“file”属性指定了包含构建数的文件名。

14.2.11 <copy>

该标记用于定义文件或文件集的拷贝，例如：

```
<copy file="myfile.java" todir="src"/>
```

在标记中可定义的属性包括：

- file，源文件
- tofile，目标文件
- todir，目标目录
- overwrite，指定是否覆盖目标文件，默认值时不覆盖
- includeEmptyDirs，指定是否拷贝空目录，默认值为拷贝

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

- failonerror，指定假如目标没有发现是否自动停止，默认值时自动停止
- verbose，指定是否列出所拷贝的文件，默认值时不拷贝

一个任务是一段可执行的代码。一个任务可以有多个属性，属性只可能包含对 property 的引用，这些引用会在任务执行前被解析。下面是任务的一般构造形式：

```
<name attribute1="value1" attribute2="value2" ... />
```

其中，name 是任务的名字；attributeN 是属性名；valueN 是属性值。

构建文件中每个任务元素可以有一个 id 属性，可以用这个 id 值引用指定的任务。这个值必须是惟一的。

每个构建文件都应包含一个清空任务，删除所有生成的文件和目录，使系统回到构建文件执行前的初始状态。执行清空任务后还存在的文件应处在版本控制系统的管理下。

14.3 Ant 数据类型*

在构建文件中为了标识文件或文件组等，经常需要使用数据类型。数据类型包含在 org.apache.tool.ant.types 包中。下面将简单介绍构建文件中可用的一些数据类型。

14.3.1 argument

对于由一个 Ant 构建文件调用的程序，通过<arg>元素，可以向其传递命令行参数。<arg>的属性包括下面的一个或多个：

- file，作为一个参数的文件名
- line，用空格分隔的多个参数的列表
- path，路径
- 如果参数中有空格，又想将它作为单独的一个值，则使用此属性。需要有一个以上属性

14.3.2 environment

对于由一个 Ant 构建文件调用的外部命令或程序，<env>元素指定向其传递的环境变量。<env>元素的属性包括：

- file，作为环境变量值的文件名。此文件名将转换为一个绝对路径
- key，环境变量名
- path，作为环境变量的路径。Ant 会将它转换为一个本地约定
- value，环境变量的一个直接量值

* 本小节主要参阅的资料见书后的参考文献（55）。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

14.3.3 filelist

定义一个文件的命名列表，这些文件无须确实存在。其可接受的属性包括：

- dir，用于计算绝对文件名的目录
- files，用逗号分隔的文件名列表
- refid，对某处定义的一个<filelist>的引用

14.3.4 patternset

可以使用模式过滤文件或路径，匹配符号的含义如下：

* —— 匹配零或多个字符。

** —— 递归匹配零或多个目录。

? —— 匹配一个字符。

例如，下面的模式将获取文件名中不包含 hello 的所有 Java 文件。

```
<patternset id="javafiles">
    <include name="**/*.java"/>
    <exclude name="**/*hello*.java"/>
</patternset>
```

patternset 包含的属性主要有：

- excludes，用逗号分隔的需要排除的文件模式列表。这是对默认排除模式的补充。
- excludesfile，每行包括一个排除模式的文件的文件名。这是对默认排除模式的补充。
- includes，用逗号分隔的需要包含的文件模式列表。
- includesfile，每行包括一个包含模式的文件的文件名。

14.3.5 fileset

fileset 定义了一组文件。可以基于模式对文件进行分组。对文件分组有不同的形式，如使用属性，使用嵌套 patternset，使用 patternset 元素。例如：

```
<patternset dir="src">
    <include name="**/*.java"/>
    <exclude name="**/*hello*.java"/>
</patternset>
```

fileset 的主要属性包括：

- dir，fileset 的基目录。
- casesensitive，默认值为 true，即区分大小写，如果置为 false，则在匹配文件名时，fileset 不区分大小写。
- defaultexcludes，确定是否采用默认的排除模式，默认值为 true。
- excludes，用逗号分隔的需要排除的文件模式列表。这是对默认排除模式的补充。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

- excludesfile，每行包括一个排除模式的文件的文件名。这是对默认排除模式的补充。
- includes，用逗号分隔的需要包含的文件模式列表。
- includesfile，每行包括一个包含模式的文件的文件名。

14.3.6 filterset

定义一组过滤器。这些过滤器将在文件移动或复制时完成文件中的文本替换。其主要属性有：

- begintoken，对于嵌套过滤器所搜索的记号，这是标识其开始的字符串，默认为@。
- endtoken，对于嵌套过滤器所搜索的记号，这是标识其结束的字符串，默认为@。
- id，对此过滤器的惟一标识符。
- refid，对构建文件中某处定义的一个过滤器的引用。

14.3.7 path

以某种在不同操作系统间可移植的方式指定路径。其主要属性包括：

- location，表示一个文件或目录。Ant 在内部将此扩展为一个绝对文件名。
- path，一个文件或路径名列表，并以分号或冒号分隔。
- refid，对当前构建文件中某处定义的一个 path 的引用。

14.3.8 mapper

定义一组输入文件和一组输出文件间的关系。其属性包括：

- classname，实现 mapper 的类的类名。当内置 mapper 不满足要求时，用于创建定制的 mapper。
- classpath，查找一个定制 mapper 时所用的类路径。
- classpathref，对某处定义的一个类路径的引用。
- from，此属性的含义取决于所用的 mapper。
- refid，对另一个 mapper 的引用。
- to，此属性的含义取决于所用的 mapper。
- type，取值为 identity、flatten、glob、merge 或 regexp 的其中之一。它定义了要使用的内置 mapper 的类型。

有关数据类型的更详细的相关信息，请参阅参考文献（55）。

14.4 Ant 的运行

运行 Ant 非常简单，在正确地安装 Ant 后，只要输入“ant”就可以了。若没有指定

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

任何参数，Ant 会在当前目录下查询 build.xml 文件。如果找到了就用该文件作为构建文件。如果使用了“-find”选项，Ant 就会在上级目录中寻找构建文件，直至到达文件系统的根。要想让 Ant 使用其他的构建文件，可以用参数“-buildfile file”，这里 file 指定了想使用的构建文件，例如：

```
ant
```

使用当前目录下的 build.xml 运行 Ant，执行默认的目标。

```
ant -buildfile test.xml
```

使用当前目录下的 test.xml 运行 Ant，执行默认的目标。

```
ant -buildfile test.xml dist
```

使用当前目录下的 test.xml 运行 Ant，执行一个叫做 dist 的目标。

14.5 Eclipse 和 Ant 的集成

Ant 与 Eclipse 的外部工具框架进行了集成。在 Eclipse 中无须安装 Ant 即可创建、编辑、配置和重用 Ant 构建文件，即在 Eclipse 集成开发环境 IDE 中直接编辑、运行 Ant。

14.5.1 创建 Ant 构建文件

Ant 支持用户在 Eclipse“工作台”中创建和运行 Ant 构建文件。这些 Ant 构建文件可以对文件系统中的资源以及工作区中的资源进行操作。

Ant 构建文件为文本文件，在 Eclipse 中创建 Ant 构建文件的最简单的步骤如下。

① 在菜单中选择“文件”→“新建”→“文件”。

② 输入文件名，文件的扩展名必须为.xml，项目构建文件的默认名通常取为 build.xml。

③ 单击“完成”按钮。

完成以上步骤后，项目中将自动生成构建文件。如构建文件名取为 build.xml，则 Eclipse 工作台将使用 Eclipse Ant 编辑器打开构建文件（空白文件），如图 14.2 所示。

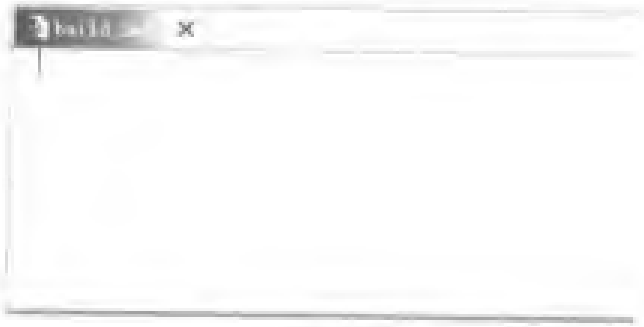


图 14.2 打开的空白构建文件

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

在 Eclipse Ant 编辑器中，构建文件的左上角将有一个“蚂蚁”图标，表明该文件为 Ant 构建文件。

14.5.2 编辑 Ant 构建文件

因为 Ant 构建文件是简单的文本文件，所以可以使用任何文本编辑器来编辑它们，但最好使用 Eclipse Ant 编辑器，因为 Eclipse Ant 编辑器对于编辑 Ant 构建文件提供了一些辅助功能，如语法着色、内容辅助和大纲视图等。

虽然 build.xml 文件的默认编辑器是 Eclipse Ant 编辑器，但是一般的.xml 文件的默认编辑器是一个简单文本编辑器，所以打开构建文件的默认编辑器有可能不是 Eclipse Ant 编辑器。

如果构建文件的默认编辑器不是 Eclipse Ant 编辑器，可以在构建文件的上下文菜单中选择“打开方式”→“Ant 编辑器”，如图 14.3 所示，从而使用 Ant 编辑器打开构建文件。设置构建文件的默认编辑器的另一种方式是通过“窗口”→“首选项”→“工作台”→“文件关联”更改默认编辑器的设置。

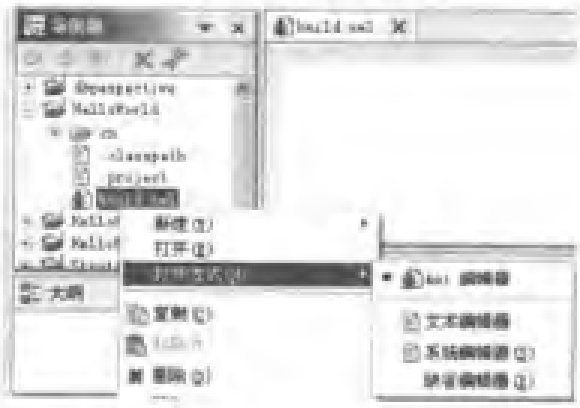


图 14.3 选择构建文件的默认编辑器

14.5.3 运行 Ant 构建文件

在 Eclipse 工作台中运行 Ant 构建文件的步骤如下。

① 在导航器视图中，选择相应的构建文件。

② 在构建文件的弹出菜单中，选择“运行 Ant”。

③ 如果选择的 XML 文件为有效的 Ant 脚本，则将打开 Ant 向导，如图 14.4 所示。

④ 在“目标”选项卡下，可从“选择要执行的目标”列表中选择一个或多个目标。选择项的次序就是运行它们的次序。次序由目标左边的方括号中的数字表示。

⑤ 开发者可选择在“主要”选项卡下的“自变量”字段中输入任何必需的自变量。

⑥ 如果想要脚本的输出显示在 Eclipse 工作台中的控制台视图中，则选择在控制台中显示执行日志。如果选择了该选项，则控制台视图将在脚本运行后显示执行结果。可

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

选择以结构化格式显示 Ant 输出。如果在日志控制台中启用了输出结构树，则目标和任务显示在一个树层次结构中。

⑦ 单击“完成”按钮。



图 14.4 启动构建文件

一旦通过 Ant 向导运行了脚本，则 Eclipse 自动将其配置为外部工具。要更改配置或重新运行脚本，可选择菜单“运行”→“外部工具”，如图 14.5 所示。



图 14.5 外部工具

14.5.4 保存和重用 Ant 选项

当运行构建文件时，目标的选项以及“运行 Ant”对话框中的所有其他选项都保存在“启动配置”中。启动配置包含按特定方式运行单个 Ant 构建文件所必需的所有详细信息。单个 Ant 构建文件可以与多个启动配置相关联。通过简单地指定相应的启动配置就可以在任一配置中快速运行 Ant 构建文件。步骤如下。

① 从 Eclipse 工作台的工具栏中的“外部工具”下拉菜单中，选择“外部工具…”。

② 此时将打开“外部工具”对话框窗口。在左边的树中选择运行 build.xml 构建文件时创建的启动配置，则右边的选项卡显示此启动配置的选项。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

③ 在该对话框的顶部，将名称进行更改，并单击“应用”按钮更改名称。

④ 在左边的树中，显示有关选择的启动配置的上下文菜单，并选择“重复”。创建构建文件的启动配置的副本，将“(1)”追加至该名称，并在树中选择新的启动配置。

⑤ 将新的配置重命名。

⑥ 在“目标”选项卡中，单击“排序…”按钮，可更改目标的次序，然后单击“应用”按钮。

⑦ 单击“运行”按钮。

此时，将运行新的构建文件，并将其输出发送至“控制台”视图。返回工具栏中的“外部工具”下拉菜单，可以看到，在历史记录中出现了新增的构建文件的条目，一个用于 Hello World，另一个用于 World Hello。只在历史记录中选择此启动配置，就可重新运行该 Ant 构建文件。

14.5.5 Eclipse 中的 Ant 视图

Eclipse 提供了使用 Ant 构建文件的 Ant 视图。在 Ant 视图中可查看 Ant 构建文件的位置，这将使得在构建文件内执行构建文件或给定的目标变得更为容易。可以将文件添加至该视图并展开构建文件以显示在其中定义的目标。此视图是树结构的，将 Ant 构建文件作为顶级条目显示，而目标和子目标则作为子代。

在 Ant 视图的弹出菜单中，可对构建文件打开编辑器，执行所选的目标或构建文件以及配置在执行时使用的启动配置。

与工作台中的其他视图相似，Ant 视图具有它自己的工具栏，如图 14.6 所示。使用工具栏中的工具按钮可将构建文件添加至该视图，并执行这些构建文件。这些按钮的主要功能如下：

- 添加构建文件

此命令在工作区中选择要添加至视图的构建文件。

- 使用搜索添加构建文件

此命令搜索构建文件，并将它们添加至 Ant 视图。可以指定要搜索的文件名模式并限制搜索的范围。

- 运行所选的构建文件的默认目标

此命令运行所选构建文件的默认目标或所选的目标。使用与构建文件相关联的启动配置执行该构建文件。这将基于视图中的选择执行目标，因此将忽略启动配置中选择目标。

- 除去所选的构建文件

此命令从视图中除去所选的构建文件。

- 除去所有构建文件

此命令从视图中除去所有构建文件。

下面将介绍在 Ant 视图中添加、编辑、除去构建文件的用法示例：

① 在工作台菜单中选择“窗口”→“显示视图”→“其他…”→“Ant”→“Ant”，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

将打开 Ant 视图。

② 默认情况下，Ant 视图为空，如图 14.6 所示。有下列两种方法可将 Ant 构建文件添加至此视图：

a. 单击 Ant 视图的窗口工具栏上的“添加构建文件”按钮。这将显示一个对话框，可在其中显式选择想要添加的那些 Ant 构建文件。

b. 单击 Ant 视图的窗口工具栏上的“使用搜索添加构建文件”按钮。这将显示一个搜索对话框，在其中可为 Ant 构建文件指定文件名模式并在整个工作区或指定工作集中进行搜索。



图 14.6 空的 Ant 视图

③ 一旦添加至 Ant 视图，Ant 构建文件条目在工作台调用期间保留在视图中直到被显式除去为止。

④ 在文件的上下文菜单中选择“打开方式”→“Ant 编辑器”，可对构建文件进行编辑。

⑤ 在上下文菜单中选择“属性…”，可编辑默认的启动配置。

⑥ 将出现“运行 Ant”对话框。此时可以修改构建文件在 Ant 视图中运行的方式。构建文件在 Ant 视图中运行时，将忽略目标的选项和次序。如果选择了顶级文件，则只有默认目标运行，而如果选择了目标或子目标中的一个，则只有该目标运行。不能在 Ant 视图中运行多个目标。

⑦ 若选择文件，然后单击“除去”按钮，将从视图中除去该构建文件，但不会从工作区中删除该文件。

14.5.6 修改 Ant 类路径

当使用可选或定制任务时，通常需要将其他的库添加至类路径。可以全局修改或者对每个启动配置修改 Ant 类路径。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

全局修改 Ant 类路径的步骤如下。

① 在 Eclipse 工作台的主菜单中，选择“窗口”→“首选项”，这将打开首选项窗口。

② 展开“Ant”类别并选择“运行时”。如果尚未在其中，则单击“类路径”选项卡。

③ 若要将 JAR 文件添加至类路径，单击“添加 Jar”按钮，并选择 JAR 文件。

④ 若要将文件夹添加至类路径，单击“添加文件夹”按钮，并选择文件夹。

⑤ 若要从类路径除去项，选择它并单击“除去”按钮。

⑥ 若要将类路径恢复至默认值，单击“恢复缺省值”按钮。

修改启动配置 Ant 类路径的步骤如下。

① 在 Eclipse 工作台的主菜单中，选择“运行”→“外部工具”→“外部工具…”。

② 选择想要修改其类路径的 Ant 配置。

③ 选择“类路径”选项卡。

④ 取消选择该选项以使用在首选项中指定的全局类路径。

⑤ 若要将 JAR 文件添加至类路径，单击“添加 Jar”按钮，并选择 JAR 文件。

⑥ 若要将文件夹添加至类路径，单击“添加文件夹”按钮，并选择文件夹。

⑦ 若要从类路径除去项，选择它并单击“除去”按钮。

14.5.7 使用其他版本的 Ant

Eclipse 平台提供 Ant 1.5.2 作为插件库。当在工作台中运行 Ant 构建文件时，默认使用版本 1.5.2。也可使用其他版本，使用其他版本的 Ant 的方式主要有两种。

（1）更改 Ant 类路径

当 Ant 运行构建文件时，它将在 Ant 类路径上查找必需的类。Ant 类路径由添加新任务、类型或库的插件组成，以及在外部工具首选项中定义的类路径。与 Ant 1.5.2 相关的两个 JAR 是 ant.jar 和 optional.jar。除去这些 JAR 并添加来自需要的 Ant 版本的 JAR。

更改 Ant 类路径后，所有以后的 Ant 构建都将使用已更新的版本，而不是默认版本。要将 Ant 类路径恢复至它的原始状态，单击属性页面上的“恢复缺省值”按钮。

（2）将 Ant 用做外部工具

如果只想测试 Ant 的较新版本或测试版本时，将其用做外部工具是一个较好的解决方案。通常当它在 Eclipse 工作台中运行时，Ant 脚本本身被认为是外部工具。将二进制 Ant 版本作为外部工具安装的步骤如下。

① 从 http://ant.apache.org 下载并安装 Ant 的二进制版本。

② 在 Eclipse 工作台主菜单中选择“运行”→“外部工具”→“外部工具…”。

③ 单击“程序”按钮。

④ 单击“新建”按钮。

⑤ 输入外部工具的名称，例如，External Ant。

⑥ 在“位置”字段旁，单击“浏览文件系统”按钮。

⑦ 查找并选择称为 ant.bat 的文件，该文件通常在 Ant 安装目录下的 bin 文件夹中。

⑧ 在“自变量”字段中，输入通常为在“工作台”外部运行脚本而输入的脚本的自

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

变量。

⑨ 在“工作目录”字段中输入构建文件的目录。

⑩ 单击“运行”按钮，执行该构建文件。

当将 Ant 作为外部工具运行时，Eclipse 添加的任何一个任务或类型都不起作用。而且，Ant 类路径首选项对于脚本的执行将不起作用。

14.6 在 Eclipse 中使用 Ant 的示例

在 Eclipse 中，Ant 构建文件可用于：

- 部署 Eclipse 插件
- 构建项目

我们将以 Ant 在 Eclipse 作为插件部署工具为例，介绍其相关应用，如为插件构建 JAR 文件并将它们作为 Eclipse 安装的一部分进行测试。

为了进行示例，首先按 10.3.4 节中所介绍的方式，使用代码生成向导创建插件项目，并选择“Hello，World”类型。在创建好该插件项目后，就可以为它创建 build.xml 文件来管理插件部署。创建项目的 Ant 构建文件的步骤如 14.5.1 一节所述。需要注意的是，在这里需要在项目下选择文件“plugin.xml”并从上下文菜单中选择“创建 Ant 构建文件”。

在“导航器”中通过双击 build.xml 文件来对该文件打开 Ant 编辑器，如图 14.7 所示。对于此文件，Ant 编辑器是默认编辑器。因为在默认情况下，在工作台首选项中存在文件关联，它将名为“build.xml”的文件与 Ant 编辑器相关联。



图 14.7 插件的构建文件

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

build.xml 文件是带有许多对部署插件有用的、目标的默认 Ant 构建文件。例如，有些目标用来为插件构建所有必需的 JAR 文件，创建包含插件中的全部内容的.zip 文件，以及清理可能创建的任何.zip 文件等。

生成 build.xml 文件后，就可使用它来为插件构建.jar 文件，其步骤如下。

① 在“导航器”视图中选择 build.xml 文件，并从上下文菜单中选择“运行 Ant…”。

② 在“运行 Ant”对话框中，将可看到“目标”选项卡。确保“build.jars”是所选列表中的惟一项。

③ 切换至“刷新”选项卡。此选项卡上的选项控制在构建文件完成执行之后工作区是否以及如何查找新的、已更改的和已删除的资源。因为刷新是成本较高的操作，所以应刷新构建文件必需的工作区的最小子集。

④ 选择“在运行工具之后刷新资源”，然后在列表中选择${project}。

⑤ 单击“运行”按钮。

⑥ Ant 构建文件运行，执行 build.jars 目标。在“控制台”视图中可见构建文件的输出。

⑦ 当构建文件完成时，注意在包含构成插件的两个类文件的项目中有一个新的 .jar 文件。如果已选择不刷新该项目，则仍将会创建.jar 文件，但是在“导航器”视图中将看不到它。

至此，已经对 Ant 构建文件 build.xml 运行了默认目标。在“大纲”视图中将可看到，默认的 Ant 构建文件 build.xml 具有若干目标和子目标。下列是一些主要的顶级目标：

- build.update.jar：此目标为插件构建 JAR，然后将它们压缩为可供“Eclipse 更新管理器”使用的适当形式。
- HelloWorldPlugin.jar：此目标为插件执行构建 JAR 的工作。
- build.jars：这是在前面运行的默认目标。它遵循 HelloWorldPlugin.jar 目标来执行其工作。
- clean：此目标删除可能已由此构建文件中的其他目标创建的所有 zip、jar 和临时目录。
- zip.plugin 此目标构建可执行文件和源 JAR，然后将所有内容压缩到单个归档中。

此 build.xml 文件只是 Eclipse 插件的默认部署构建文件。有时可能需要修改此构建文件以处理特定项目。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

第 15 章　Eclipse 环境下的 MySQL 等数据库应用开发

MySQL 是一个多用户的、多线程的关系数据库管理系统。本章将介绍 Eclipse 环境下的 MySQL 数据库应用开发，包括 MySQL 的安装和配置、数据源的配置，以及如何使用 JDBC 访问 MySQL 等。

15.1　MySQL 简介

MySQL 是一个精巧的 SQL 数据库管理系统，虽然它不是开放源代码的产品，但在许多情况下可以自由使用。由于它的强大功能、灵活性、丰富的应用编程接口及精巧的系统结构，得到了广泛的应用。

MySQL 整套软件包括一个 SQL 服务器、客户端程序、管理工具以及应用编程接口。MySQL 可以运行在多种操作系统之上，如 UNIX、Linux、Windows 和 OS2 等。

MySQL 可以被自由地使用，包括被商业实体的评估或不受支持的内部使用。然而，MySQL 的商业目的分发，或任何部分或全部包含或从 MySQL 派生的东西，都需要申请一个书面的商业许可证。

15.1.1　MySQL 的获取和安装

从MySQL 的 Web 站点下载 MySQL 数据库，下载的 URL 为：http://dev.mysql.com/downloads/。MySQL 目前最高的发布版本是 4.0，但也可下载 4.1 的 a 版本。针对 Windows 操作系统的 MySQL 4.0 的下载文件是 mysql-4.0.20c-win.zip。将该压缩文件解压到任一目录中，即完成 MySQL 的安装。Windows 版本 MySQL 也有安装形式的（即带有安装程序）。

此外，为了将 Eclipse 中开发的应用程序连接到 MySQL 数据库，需要从 MySQL 的 Web 站点下载 JDBC driver for MySQL，下载的 URL 为：http://dev.mysql.com/downloads/connector/j/3.0.html。目前，MySQL 的 JDBC 的驱动程序的最高版本为 3.0.14，相应的压缩文件名为 mysql-connector-java-3.0.14-production.zip。将该压缩文件解压到任一目录中。

15.1.2　MySQL 的配置

MySQL 安装后有一个初始用户账号 root，它的初始口令为空。MySQL 的安全机制

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

要求用户为 root 账号重新设置口令，步骤如下。

① 启动 MySQL 服务器。

② 在 DOS 命令行中转入 MySQL 安装路经的 bin 子目录。

③ 执行命令：mysql，这将调用 mysql 客户端程序。

④ 执行 MySQL 数据库命令：use mysql。

⑤ 为 root 账号重新设置口令，假设新口令为：root123，执行 mysql 命令：

```
UPDATE user SET Password=PASSWORD('root123') WHERE user='root';
FLUSH PRIVILEGES;
```

⑥ 执行 MySQL 命令：exit，退出数据库。

15.2 数据库连接

Java 提供了大量用于操作数据库、执行 SQL 语句的接口和对象。通常使用 JDBC API 连接数据库和执行 SQL 语句。JDBC 是一种规范，它让各数据库厂商为 Java 程序员提供标准的数据库访问类和接口。JDBC 为 Java 程序开发人员访问关系型数据库提供了一致的机制。使用 JDBC，Java 程序开发人员无须关注不同数据库管理器具体实现的细微差别。

15.2.1 JDBC 对象

Eclipse、Tomcat 和 MySQL 都支持 JDBC 2.0。JDBC 2.0 API 可以在下面两个包中找到：

- Java.sql，JDBC 2.0 核心 API
- Javax.sql，JDBC 2.0 标准扩展 API

Java 提供了不同的程序接口用于连接数据库和执行 SQL 语句。Java API 中的接口包括：

- CallableStatement

CallableStatement 接口用于执行返回 OUT 参数值的存储过程。CallableStatement 对象是 PreparedStatement 的子类，除继承了 PreparedStatement 的全部方法外，又另加入了注册 OUT 参数和获取存储过程的返回参数的不同方法。

- Connection

Connection 接口对象用于 Java 应用程序与数据库的连接。它可用于建立各种执行 SQL 语句和存储过程的 Statement 对象。同时，Connection 对象中也可设置事务处理的连接参数。

- DatabaseMetaData

DatabaseMetaData 接口提供了各种获取数据库信息的方法。该接口可用于获取特定的数据库信息，数据库中表的关键字、列等信息。

- Driver

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

Driver 接口对象是由 JDBC 供应商提供的与特定数据库相关的 Driver 对象，所有的 JDBC 驱动程序都必须实现 Driver 接口。

• PreparedStatement

PreparedStatement 接口用于执行动态的 SQL 语句和存储过程。动态 SQL 语句中的值在执行时才可确定，在创建语句时是未知的。如在数据库应用中，同样的语句可能执行多次，然而每次的参数不同。

• ResultSet 接口

ResultSet 接口对象用于接收检索语句（select）返回的结果。ResultSet 接口对象通过游标查看 select 语句的结果集。

• ResultMetaData

ResultMetaData 用于获取返回的结果集的有关信息。利用 ResultMetaData 接口可以得知结果集中列的数目、列名和类型，以及关于返回的 ResultSet 的信息。

• Statement

Statement 用于执行通常的静态 SQL 语句。它有两个最主要的方法 executeUpdate(String sql)方法和 executeQuery(String sql)方法。对于 INSERT、UPDATE 和 DELETE 语句，使用 executeUpdate(String sql)方法；对于 SELECT 语句，调用 executeQuery(String sql)方法，并返回 ResultSet 对象。

15.2.2 JDBC 驱动程序的类型

JDBC API 并不能直接访问数据库，它需要通过数据库厂商提供的 JDBC 驱动程序才能访问数据库。因此编写访问数据库的 Java 程序时，必须把特定数据库的 JDBC 驱动程序包加到类路径中。JDBC 驱动程序总共有四种类型：

① JDBC-ODBC 桥

② 本机 API，部分是 Java 的驱动程序

③ 面向数据库中间件的纯 Java 驱动程序

④ 直接面向数据库的纯 Java 驱动程序

第一类 JDBC 驱动程序是 JDBC-ODBC 桥再加上一个 ODBC 驱动程序。由于 ODBC 会在客户端装载二进制代码和数据库客户端代码，这种技术不适用于高事务性的环境。此外，这种方式不支持完整的 Java 命令集，而局限于 ODBC 驱动程序的功能。

第二类 JDBC 驱动程序是本机 API 的部分 Java 代码的驱动程序，用于把 JDBC 调用转换成主流数据库 API 的本机调用。这类驱动程序也存在与第一类驱动程序一样的性能问题，即客户端载入二进制代码的问题，而且它们被绑定了特定的平台。使用这些驱动程序的开发者必须使用不同数据库厂商针对不同操作系统发行的各个驱动程序版本。

第三类 JDBC 驱动程序是面向数据库中间件的纯 Java 驱动程序，JDBC 调用被转换成一种中间件厂商的协议，中间件再把这些调用转换到数据库 API。这类 JDBC 驱动程序不再需要客户端的本机代码，因此速度比前两类快。另外，开发者还可以利用单一的驱动程序连接到多种数据库。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

第四类 JDBC 驱动程序是直接面向数据库的纯 Java 驱动程序，即“瘦”驱动程序，它把 JDBC 调用转换成某种直接可被 DBMS 使用的网络协议。这样，客户机和应用服务器可以直接调用 DBMS 服务器。对于第四类驱动程序，不同的 DBMS 其驱动程序也不同。因此，在一个异构计算环境中，驱动程序的数量可能会比较多。但是，此类驱动程序性能最强。

15.2.3 JDBC 的使用

使用 JDBC 2.0 核心 API 包中的类型，可以完成所有基本的数据库操作。下面将简述使用 JDBC 操纵数据库的具体步骤，如获取驱动程序管理器连接、使用连接来创建语句、执行语句、返回操纵结果，以及最终断开连接等。

1. 加载合适的 JDBC 驱动程序

为了与特定的数据源连接，JDBC 必须加载相应的驱动程序。在应用程序中，JDBC 驱动程序只须加载一次。

加载不同类型的 JDBC 的语句并不相同。如加载 JDBC-ODBC 桥类型的 JDBC 驱动程序则通过语句 Class.forName("Driver Name")，例如：

```
Class.forName("com.mysql.jdbc.Driver");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver")
Class.forName("oracle.jdbc.driver.OracleDriver")
```

第四种类型的 Oracle 数据库的 JDBC 驱动程序的加载语句为：

```
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
```

DB2 JDBC 驱动程序在安装 DB2 客户端或服务器时安装。DB2 提供了一个 app 驱动程序：COM.ibm.db2.jdbc.app.DB2Driver。运行在安装了 DB2（客户端或服务器）的机器上的应用程序使用这个驱动程序。相应的加载语句为：

```
DriverManager.registerDriver(new COM.ibm.db2.jdbc.app.DB2Driver());
```

DB2 还提供了一个网络驱动程序：COM.ibm.db2.jdbc.net.DB2Driver。当 Java 程序运行在没有 DB2 安装的机器上时，使用这个驱动程序。驱动程序的类文件打包在文件 db2java.zip 中。

2. 从驱动管理器获得连接

使用代表数据库 URL 的字符串，从驱动管理器获得连接。对一些数据库连接而言，需要合法的用户 ID 和密码。URL 的格式视使用的驱动程序而定。对于 URL 的正确格式以及所使用的驱动程序，可参考相应的数据库的文档。

例如，Oracle 的连接语句类似：

```
    Connection dbConn = DriverManager.getConnection ("jdbc:oracle:thin:
@127.0.0.1:1521: databasename", "username", "password" );
```

在上面的语句中 127.0.0.1 指的是本机，其值可以为数据库所在的计算机的 IP 地址；

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

1521 是 Oracle 数据库的默认侦听端口。“username”和“password”，对应数据库连接的用户名和口令。

同样地，DB2 的连接语句类似：

```
Connection dbConn = DriverManager.getConnection ("jdbc:db2://ServerIP:50000/
databasename, "username", "password");
```

在上面的语句中，ServerIP 指 DB2 数据库驻留的计算机；5000 是 DB2 默认的侦听端口。databasename 是所要连接的数据库名。

3. 基于连接创建语句

在 JDBC 应用程序中，若要执行 SQL 语句，首先需要使用 Connection 对象的 createStatement()方法创建一个 Statement 对象，其语句如下：

```
Statement sqlStatement = connection.createStatement();
```

利用 Statement 对象可以执行静态 SQL 语句。静态 SQL 可以是 Select 语句、Delete 语句、Update 语句和 Insert 语句，也可以是存储过程的调用语句。

也可以使用连接的方法 prepareStatement()和 prepareCall()，分别获得预处理语句及调用存储在数据库的存储过程。

除了执行存储过程，PreparedStatement 接口还可用于执行动态 SQL 语句。创建用于 PreparedStatement 对象的动态 SQL 语句时，用“？”作为所有动态参数的占位符。例如，下面是利用 PreparedStatement 对象查找 student 表中具有指定学号的学生信息的语句：

```
String sql = "select * from student where s# = ?";
PreparedStatement prepare = connection.prepareStatement(sql);
```

在这里，SQL 语句中所用的学号的具体值用“？”代替，从而给出不同的参数就可以多次执行同一条动态 SQL 语句。

4. 使用语句对象实例执行数据库操作

Statement 接口提供了许多创建和执行 SQL 语句的方法。可使用语句对象实例来执行查询，实际的 SQL 语句是作为参数传递的。其中，executeQuery()执行 SQL Select 语句，返回包含满足指定 SQL 语句条件的记录组成的结果集；executeUpdate()执行 SQL 的更新语句，包括 Update 语句、Delete 语句和 Insert 语句。

下面是在 JDBC 程序中，将 student 表中学号为“a123”的学生的姓名改为“张三”的代码：

```
String sql = "update student set name = \'张三\' where s# = \'a123\'"
int recordNumber = sqlStatement.executeUpdate(sql);
```

注意，在上面的代码中，我们在单引号前使用了转义字符“\”，主要是为了避免 JDBC 将指定字符串前的单引号解释为 SQL 命令的结束字符。返回值 recordNumber 表示了 update 语句涉及的记录数。

又如，在程序中查询 student 表中学生的学号和姓名的代码如下：

```
String sql ="select s#, name from student";
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
ResultSet results = sqlStatement.executeQuery(sql);
```

5. 处理执行的返回结果

在上面的例子中可以看到，executeUpdate()返回一个整数值，表示所影响的数据库中的记录的记录数。检索操作 executeQuery()返回类型为 ResultSet 的结果集。

返回的结果集有一个游标（cursor，或称“光标”），可以进行遍历。首先，结果集游标指向一个虚拟的行，它恰好位于返回的第一行之前。Next()方法把指针移到第一行，如果有第一行则返回真，如果没有则返回假。下面是处理上面所执行的检索操作的返回结果的代码示例：

```
while (results.next()) {
      String resultDisplay = "";
      resultDisplay = resultDisplay + results.getString ("s#");
      resultDisplay = resultDisplay + " ," + results.getString("name");
      resultDisplay= resultDisplay + "  \n" ;
      System.out.println(resultDisplay);
}
```

在 while 循环中，可从当前行中提取列值。为获取一个列可能包含的不同数据类型，有很多种的 get 方法，如 getString()、getInt()和 getFloat()等。必须使用正确的方法，这意味着必须清楚数据库的结构，如传递一个字符串来代表列名。传递到这些方法的参数也可以是列的序号。这是从 1 开始的号码，如第一列是 column 1。

6. 不再需要连接时，关闭连接

close()方法显式地关闭 Statement 对象。关闭 Statement 对象将释放相关的资源。调用 close()方法的语法如下：

```
connection.close();
```

15.2.4 MySQL 数据连接池

JDBC 2.0 提供了 javax.sql.DataSource 接口，它可建立与数据库的连接。在 Java 应用程序中访问数据库时无须编写连接数据库的代码，可以直接从数据源获得数据库连接。

JDBC 2.0 的标准扩展 API 通过数据源和连接池，可以实现高效的数据库访问。连接由连接池共享。Java 程序访问数据库时，只须从连接池中取出空闲状态的数据库连接。当应用程序访问数据库结束，再将连接送还连接池。

之所以使用连接池，主要因为建立和销毁一个数据库连接是一种很费时间的操作，尤其是在大多数情况下，连接的使用时间很短，而连接数较多。

因为 Tomcat 4.1 才开始内置对数据库连接池的支持，如果需要使用数据库连接池，一定要使用 Tomcat 4.1 以上版本。版本 4.1.18 后，内置的连接池为 DBCP，DBCP 也是由 Jakarta Commons 的一个子项目，其配置示例如下：

第一步：配置 server.xml

在 server.xml 文件中找到：

```
<!-- Tomcat Root Context -->
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理


```
<!--
    <Context path="" docBase="ROOT" debug="0"/>
-->
```

将其改为：

```
<!-- Tomcat Root Context -->
  <Context path="" docBase="ROOT" debug="0">
  <Resource name="jdbc/MySQLDB" auth="Container" type="javax.sql.
  DataSource"/>
  <ResourceParams name="jdbc/MySqlDB">
   <parameter>
     <name>factory</name>
     <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
   </parameter>
   <parameter>
     <name>driverClassName</name>
     <value>com.mysql.jdbc.Driver </value>
   </parameter>
   <parameter>
     <name>url</name>
     <value> jdbc:mysql://DB_SERVER/ MySQLDB </value>
   </parameter>
   <parameter>
     <name>username</name>
     <value>admin</value>
   </parameter>
   <parameter>
     <name>password</name>
     <value>admin123</value>
   </parameter>
   <parameter>
     <name>maxActive</name>
     <value>20</value>
   </parameter>
   <parameter>
     <name>maxIdle</name>
     <value>10</value>
   </parameter>
   <parameter>
     <name>maxWait</name>
     <value>-1</value>
  </parameter>
 </ResourceParams>
</Context>
```

在上述配置中，factory 参数基于对象工厂，这里取值为 org.apache.commons.dbcp。BasicDataSourceFactory，即 DBCP 自带的工厂，也可以用别的。此外，username 和 password 为连接数据库的用户名和口令。在本例中，假设分别为“admin”和“admin123”。maxActive 是最大激活连接数，这里取值 20 个，表示同时最多有 20 个与数据库的连接。maxIdle 是最大空闲连接数，这里取值 10 个，表示即使没有连接请求时，依然可以保持 10 个空闲的连接，而不被清除，随时处于待命状态。maxWait 是最大等待秒钟数，这里取值-1，表示无限等待，直到超时为止，也可以取其他值，如 5000，即表示 5s 后超时。

第二步：配置 web.xml

打开 webapps/ROOT/WEB-INF 下的 web.xml，加入如下内容：

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

```
<resource-ref> <description>MySQL Datasource </description>
    <res-ref-name>jdbc/OracleDB</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
```

第三步：拷贝驱动程序 JAR 文件

将 MySQL 的 JDBC 驱动程序的 JAR 文件拷贝到 Tomcat 安装目录的 common/lib 下。

15.2.5 一个简单的数据库查询的例子

在前面的小节中，我们已经介绍了数据库应用程序开发的相关内容。下面我们将给出一个在 Eclipse Java 程序中通过 JDBC 访问 MySQL 数据库，查询并显示 student 表中学生的学号和姓名的一个简单的数据库查询的代码。

```
import java.sql.*;

public class DataAccess {

    public static void main(String[] args) {
        try {

            Class.forName("com.mysql.jdbc.Driver");
            String url =
"jdbc:mysql://localhost:3306/sms_db?useUnicode=true&amp;characterEncoding=
GB2312";
            Connection connection = DriverManager.getConnection(url);
            Statement sqlStatement = connection.createStatement();
            String sql = "select s#, name from student";
            ResultSet results = sqlStatement.executeQuery(sql);

            while (results.next()) {
                String resultDisplay = "";
                resultDisplay = resultDisplay + results.getString("s#");
                resultDisplay =
                    resultDisplay + " ," + results.getString("name");
                resultDisplay = resultDisplay + "  \n";
                System.out.println(resultDisplay);
            }

            connection.close();

        } catch (SQLException e) {
        } catch (Exception e) {
        }

    }
}
```


更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

附录 A　开放源代码软件

开放源代码软件已经获取了越来越多的注意，并且由于 Linux、Apache、Eclipse 等相继获取的巨大成功，开放源代码运动将给 IT 产业带来革命性的变化，开放源代码软件在软件业，甚至在整个 IT 业界扮演了越来越重要的作用。许多用户对于开放源代码软件的兴趣越来越大，开放源代码给了用户多样化的选择。

由于理念冲突而长期对开放源代码运动含有敌意，曾经将开放源代码许可方式比做癌症的软件业巨人 Microsoft 也对开放源代码软件进行秘密研究，并宣称“喜欢上”开放源代码的概念，将部分软件程序代码开放给大众，并鼓励大众参与进行修正与开发。盖茨本人也宣布微软将开放 Windows 操作系统的程序代码给部分大客户。如微软已经向开发商以共享源代码的许可方式开放了其 Windows CE 嵌入式操作系统的源代码。

开放源代码方案允许任何团体或个人参与个人或者团体公开的技术的开发和商业操作。现在，开放源代码运动的兴起，几乎是爆炸性地发展。如开放源代码软件，开放源代码模式，开放源代码文化，开放性文档的撰写。目前公开源代码已经发展到超出了技术和商业的领域。在哈佛大学法学院，拉瑞·莱锡格教授和查尔斯·尼森教授已经将公开源代码模型引入了法学界。

A.1　什么是开放源代码软件

按照开放源代码促进会的定义，开放源代码软件是指源代码公开的软件，用户可以获取源代码的软件，用户可免费下载软件的源代码或者以低廉的价格购买，如 Linux 操作系统、Apache 网络服务器。并且此软件的使用、修改和发行也不受许可证的限制。除了公开源代码，按照开放源代码运动的定义（最新版本 V1.9, http://opensource.org/docs/def_print.php），开放源代码的发布必须满足以下条件（注意，斜体部分不是定义的一部分，而是对定义的解释）：

1．自由地重新发布

对于包含多个不同来源的聚合软件，许可证并不限制任何当事人或组织对包含开放源码软件的聚合软件的销售或分发。作为被发布软件的组成之一的开放源代码软件的许可证对此类销售并不需要版权特许费或其他任何费用。

之所以强制要求开放源代码软件的许可证允许自由重新分发，主要是为了消除为获得短期的销售收入而放弃长期收益的诱惑。假如不坚持这一点，可能将有大量的压力促使合作者放弃承诺。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

2. 源代码

程序必须包括源代码，并且必须允许以源代码方式和已经编译的形式一起发布。如果产品的某个部分没有与源代码一同发布，那么必须提供通行的、并且不需要支付合理的复制费用之外的任何费用的手段以获得源代码，提供从因特网免费下载是一种更可取的方式。源代码必须是程序员对程序进行修改的优先形式。故意地使源代码变得含混晦涩是不允许的。也不允许给出诸如预处理器或翻译器的输出等中间结果。

程序不进行修改就无法改进，因此我们需要易于理解的源代码。因为我们的目的是为了软件易于改进，因此我们必须使软件易于修改。

3. 派生作品

许可证必须允许修改软件和派生软件，并且必须允许它们按照原软件的许可证的条款进行发布。

仅仅可以阅读源代码，并不足以支持独立的同行复查和程序的快速改进。为了加快程序的改进速度，人们需要试验和对修改进行重新发布。

4. 作者的源代码的完整性

只有在许可证允许创建程序时对程序进行修改的补丁文件和源代码一起发布时，许可证才可能限制源代码以修改形式的发布。许可证必须明确地允许由修改后的源代码生成的软件的发布。许可证可以要求派生的作品采用不同的名称或版本号以区别于原来的软件。

鼓励软件的各种改进是一件好事，然而用户有权知道谁对他们正在使用的软件负责。作者和维护者也有一个互惠的权力知道用户期待他们做什么，从而树立和维护他们的名誉。

因此，开放源代码许可证必须保证可以很容易得到源代码，但可以要求以原来的基础代码加上补丁的方式发布。按照这种方式，可以获得“非正式”的修改，但是可以很容易地将他们从基础代码中区分出来。

5. 不得歧视任何个人或团体

许可证不得歧视任何个人或者群体。

为了从该过程中获取最大的利益，应该给尽可能多的形形色色的个人和群体以同等资格对开放源代码软件做出贡献。为此，开放源代码许可证禁止把任何人排除在该过程之外。包括美国在内的一些国家，对一些类型的软件有出口限制。遵循开放源代码定义的许可证可以告诫领有许可证者一些应用限制，并且提醒他们注意遵循法律。然而许可证自身并不包含此类限制。

6. 不得歧视任何使用领域

许可证不得限制任何人在任何领域使用该开放源代码程序。例如，不得规定程序不能应用于商业领域或基因研究领域。

该条款的主要目的是禁止许可证中含有使开放源代码软件无法在商业上使用的限制。我们需要商业用户加入开放源代码社区，而不让他们感到被排除在外。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com

本教程由站长百科收集整理

7. 许可证的发布

与程序有关的权利必须适用于该程序的任何使用者，并且程序的使用者也不需要为了使用该程序而获得其他许可证的许可。

本条款的目的是禁止通过非直接的手段，例如需要签署非公开的协定，使软件无法公开。

8. 许可证不能针对于一个产品

与程序有关的权利不能由该程序是否作为某个软件产品的一部分来决定。如果程序从那个发布中被抽出来，并且按照程序的许可证的条款进行使用和发布，那么得到该程序的当事人或组织将获得与得到原程序的使用者相同的权利。

本条款用于预防其他许可证陷阱。

9. 许可证不能影响其他软件

许可证不得向与采用它的软件一同发布的其他软件提出任何限制。例如，许可证不能坚持要求在同一媒体上发布的其他程序都是开放源代码软件。

开放源代码软件的发布者有权为他们自己的软件做出选择。

通用公共许可（GPL）满足这条要求。按 GPL 仅当该软件形成一个单独的作品才继承通用公共许可，而不是所有被分发的软件。

10. 许可证在技术上应该中立

许可证条款不应断言任何单个技术或界面风格。

有些许可证只有明确地同意才能在许可证颁发者和许可证获取者之间建立合约。这个条款主要针对这类许可证。那些限制条件可能和一些重要的软件分发方法（诸如 FTP 下载、CD-ROM 的组合复制或网站镜像）冲突；它们也可能阻碍代码的复用。非网站方式的下载无法实现通过单击来直接确认合约，但软件分发机制也应支持非网站方式的分发方式。隐藏代码（或隐藏代码的可复用部分）也应能在不支持弹出式对话框的非图形用户界面环境中运行。

开放源代码软件与专有代码软件最大的差别在于它的许可协议，只要向公众开放修改后的代码，任何组织和个人都可以免费得到开放源代码软件，并对它进行自由修改。开放源码软件通常是有版权的，它的许可证可能包含这样一些限制：保护软件的开放源码状态，著者身份的公告，或者开发的控制。开放源代码产品共同使用的授权模式为通用公共许可证，参见附录 C。

A.2 使用开放源代码的优点

如果没有源代码，一方面，各种软件之间的配合经常会出现问题，从而影响系统的稳定性，而且源码分发是实现交叉平台可移植性的惟一实际可行的办法。开放源代码非常有利于形成标准，让各种软件更好地实现集成。另一方面，仅靠用户的黑盒测试，系统的漏洞可能长期威胁到用户的安全。开放源代码的软件同样存在漏洞，但开放源代码

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

最大的好处就是透明，软件开发人员和使用者可以在合法的情况下自由测试、修改，不同的开发人员可以参与整个软件体系的构建，并使之完善。

源代码对于查错和理解产品工作原理来说是很重要的。只有开放源代码，才可以让软件系统经历真正的检验，尽早发现和解决包括系统漏洞在内的各个细节方面的问题。更多的独立同行对代码和设计的双重审查，以及大部分作者对自己作品的荣誉感，都对其优良的质量有所贡献。因此，开放源代码会使系统更稳定和更安全。

在大的软件公司，只有极少数人能接触到源码，而这些能接触源码的人通常用户都无法直接接触，使得软件的开发局限于特定公司的内部，其他开发者无法利用已有的代码和软件，造成重复劳动和社会财富的极大浪费。源代码的保密使得公众无法查看和改进其安全性，从而致使许多软件存在安全隐患。拥有源代码可以尽早发现和解决包括系统漏洞在内的各个细节方面的问题，并且可以使该软件的用户控制那些他们的业务所赖以生存的工具。

开放源码给用户极大自由，由于源代码的公开，用户可以自由、方便地修改软件，使他们能够按照自己的业务需求定制软件。当用户对某以开放源代码产品不满意时，其他的开发者将使用该源代码开发新的产品以解决所存在的问题。采用开放源代码软件是一种避免被绑定在一种专有信息技术产品上的途径。用户自己将直接控制软件，而不是由软件供应商控制，这在传统私有软件模式下是难以实现的。

应用开放式源代码软件将具有较低的总体拥有成本。开放源码使得软件在版权许可方面比私有软件具有更大的灵活性，也能给用户安装软件以更大的自由度。

此外，开放源代码软件的电子邮件和 Usenet 新闻支持是非常有效的。几乎所有开发开放源代码软件的组织都提供免费的电子邮件支持，提供邮件列表和 Usenet 新闻组。由于这些机制实现了 7×24 的服务模式，因此往往比工作时间的受限服务电话等传统支持方式更加实际有效。

开放源代码运动给我们描绘了软件开发的美妙的过程：某团体或个人提供种子源代码→通过网络等技术手段提交给开放源代码社区→任何个人和团体都可修改、复制、散发→再次提交给开放源代码社区，随着这种过程的不断循环，最后由初始的种子源代码得到规模很大的很成熟的最终软件。

由于开放源代码软件通常可免费获得，这为实验性试验创造了机会。开放源代码社团可为用户开发出更多满足特别需求的软件。正是由于这一特性，全世界的编程人员都在无偿地开发开放源代码软件，并不断地解决其中出现的问题。开放源代码能够把个人的知识与他人进行分享，从而加快软件的开发速度、减少软件中的错误和缺陷、使软件更加完美，功能更为强大。

A.3 开放源代码运动面临的挑战

开放源代码运动生态链如何维持生存？企业如何营利？如何获取软件研发的后续资金？因为开放源代码软件的非独占性影响了开发者对利润的获取，所以开放源代码

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

产业的营收是一个重要的议题。尽管开放源代码对大多数人都意味着免费，但相关厂商和技术人员仍能从中获利，如：（1）提供额外的服务，通过提供认证、技术支持、培训和其他商业性服务来增加收入，并收取服务费用。（2）开发和销售在开放源代码平台上运行的专有软件，向用户提供解决方案。如 IBM 不仅销售运行 Linux 的服务器和大型主机，还提供应用软件和支持服务。

对于开发源代码软件，人们可以自由修改并公开发布，如果没有一个统一的标准，种子源代码可能会发展成为各种不同版本的最终软件。由于开放源代码使用的范围相当广泛，因此其软件在归档、支持与封装等功能方面相对薄弱，而这些缺陷将限制使用开放源代码的用户数量。

随着软件技术和实际需求的进一步发展，软件的规模越来越大，需要高度的统一和协调，而开放源代码运动的组织则相对松散，因此需要创立规范的组织体系。

开放源代码软件存在如何争取用户的问题。安装和使用开放源代码软件有时需要配置或编译源代码的能力和经验，这将提高最终用户的安装难度。由于许多用户往往有多年的使用和开放源代码软件相竞争的专有软件的经验。无疑，这将提高开放源代码软件所面临的挑战的门槛，特别是在桌面系统。因此，如何促使他们放弃多年的使用习惯，迁移到开放源代码软件产品，是开放源代码软件需要解决的问题之一。大部分开放源代码软件主要是支持 UNIX 和 Linux 系统，部分支持 Windows 平台，而事实上 Windows 操作系统尤其在桌面系统中占据了垄断地位，因此开放源代码软件需要很强的可移植性。在这一点上，开放源代码软件显然比专有软件更有优势。

此外，开放源代码运动在法律上还存在一些争议，如是否有利于知识产权的保护？是否有利于鼓励和促进技术创新？商业公司运用著作权和商业秘密两种手段对软件及其源代码以独占的方式进行开发。这种传统的软件知识产权方式已经给微软等软件公司带来了巨大的经济效益并激发了软件产业的发展。与之相比，开发源代码运动还有待进一步的考验。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

附录 B　自由软件

“开放源代码”与“自由软件”两种说法，虽然它们来自两个组织的定义，但其精神本质上是一样的，都是强调软件的“开放”与“共享”，并为方便用户使用提供了“自由空间”，但是也存在许多不同之处。自由软件并非是新生事物，其历史甚至比“版权软件”（即后面所说的“私有软件”）还要悠久，可以追溯到软件的起源阶段，自由软件的发展是包括了整个 UNIX、因特网自由软件和黑客（经典意义上的黑客，注意和骇客相区别）文化的历史，并形成了最初的软件的“知识共享”观念。

自由软件即“Free Software”，也有人将其称为免费软件（Freeware）。其实自由软件和免费软件是两回事。免费是指购买软件的费用。但可能存在缺乏许可证限制而带来的费用。免费软件的开发者仍拥有软件的版权。有几种免费的软件类型，但不一定是开放源代码软件，因为它们具有某些限制或是缺乏公开的源代码。例如：

（1）共享软件（Shareware）：一般没有技术支持、没有公开的源代码，它希望得到使用者的资助，通过网络进行分发，使用者先试用后选购。即使出于个人的非营利目的，也不允许在不支付许可费的情况下进行拷贝和分发。

（2）捆绑软件：由商业软件携带并且不用增加额外的费用，但是它一般具有很有限的使用许可，没有自由可用的源代码。如 Microsoft Internet Information Services、Windows Media Player 等。

（3）试用软件：免费或以较低费用试用的软件，是商业软件的一部分，软件在开发过程中，供用户测试及收集意见和协助清查错误，并提供预览机会。试用软件没有自由可用的源码，也没有技术支持。

（4）非商业用途使用软件：这是可以免费使用该软件，甚至包括源代码，但是同时它有严格的许可证限制它的使用。

虽然自由软件和免费软件的英文都是“Free Software”，但严格说，含义并不一样。因为自由软件的基本含义包括自由地运行、学习、研究、拷贝、改进和传播软件及其相关文档，其更主要的含义是“Freedom”，而是否免费获得并不是自由软件的必要条件，其本质是思想共享、知识共享和源码共享，开发者通过源代码的学习可以将知识和经验应用到自己的软件开发中。自由软件程序包含源代码，任何人可以自由获取和修改，但自由软件并不仅仅意味着是开放源代码软件，它的本质特征还包括：任何组织或个人都可以自由地再发布自由软件或其派生作品；不限制自由软件的应用范围，可应用于包括商业领域在内的一切范围；不影响和它一起发布的其他软件等。总之，自由是自由软件的最大特征。自由软件的主要目标是使得开发的软件是自由的，而开放源代码的主要目标是开发高质量的软件，它们之间的最基本的区别是各自的优先权不同。

GNU 工程是自由软件运动的基础，Richard Stallman 于 1983 年发表的《GNU 宣言》

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

是自由软件运动发起的标志。他明确地把自由软件和私有软件（Proprietary Software，或称做专有软件）对立起来。GNU 是自由软件运动的技术平台，自由软件的参与者通过自由软件推动自由软件运动的发展。为了和私有软件的版权（Copyright）概念相对应，他提出了新的版权概念 Copyleft（拷贝留下），鼓励对自由软件的自由复制、鼓励自由的协作精神。通用公共许可证是自由软件基金会为自由软件申请的版权。所有自由软件的作者都可以自愿把编制的软件置于许可证的保护之下，任何使用者也可以在遵循许可证的前提下成为自由软件的合法用户。

私有软件利用知识产权的保护，实现技术壁垒和市场壁垒，从而获得垄断或事实上的垄断，从而限制了软件业竞争和发展，并限制了用户的知情权和选择权。

私有软件由开发者出售拷贝并提供技术服务，用户只有使用权，但不得进行拷贝、分发和修改。而自由软件开发者往往提供软件全部源代码，任何有户都有权使用、拷贝、修改或分发该软件，同时使用者也有义务将自己修改过的程序代码公开。因此将自由软件稍加修改将之作为私有软件或将修改的部分作为私有软件都是违法的，从而保护了自由软件的自由性质及其连续性不被改变与中断。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

附录 C　通用公共许可证

传统的知识产权许可是为了剥夺其他人对软件的分享和修改权，从而维护权利人对软件成果的独占。而通用公共许可是利用现有法权，以合法方式保护自由软件的自由性质，获取通用公共许可意味对独占权利的放弃。通用公共许可保护自由软件不被某些组织或个人私有。由于对利用通用公共许可证获取的软件的修改也必须遵循通用公共许可，因此通用公共许可可保持自由软件的连续性。下面是 GPL 通用公共许可证发布版本。

GNU 通用公共许可证

第二版　1991 年 6 月

版权所有（C）1989，1991 自由软件基金会。
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA 。
允许任何人复制和发布这一许可证文档的拷贝，但不允许对它进行任何修改。

导　言

大多数软件许可证的目的是为了剥夺共享和修改软件的自由。相反，GNU 通用公共许可证力图保证共享和修改自由软件的自由——保证自由软件对所有用户是自由的。本许可证适用于绝大多数自由软件基金会的软件和那些开发者申请本许可证的程序。（其他的一些自由软件基金会软件使用库通用公共许可证。）你也可以将它用于你的程序。

当我们谈到自由软件时，我们指的是自由而不是价格。我们的 GNU 通用公共许可证目的是为了保证发布自由软件的自由（如果你愿意，你可以对此项服务收取一定的费用）；保证你能收到源程序或者在你需要时能够得到它；保证你能修改软件或将它的一部分用于新的自由软件；而且还保证你知道你能做这些事情。为了保护你的权利，我们禁止任何人不承认你的权利或者要求你放弃这些权利。如果你分发了软件的拷贝或者修改了软件，这些将转化为你的责任。例如，不管是收费还是免费，如果你分发这样一个程序的拷贝，你必须将你具有的一切权利给予你的接受者。你必须保证他们能收到或得到源程序。并且给他们看这些条款，以便他们知道这些权利。

我们采取两项措施来保护你的权利：

（1）给软件以版权保护。

（2）给你提供许可证，提供你复制、分发和 / 或修改这些软件的法律许可。

同样，为了保护作者和我们，我们需要清楚地让每个人明白，自由程序软件没有担

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com

保。如果其他一些人修改了程序，并继续加以传播。我们需要它的接受者知道：他们所得到的并不是原来的版本。任何由其他人引入的任何问题，不应损害原作者的声誉。

最后，任何自由软件不断受到软件专利的威胁。我们希望避免自由软件的再分发者个别地获得专利许可证，从而在事实上将自由软件变为私有软件这样的风险。为防止这一点，我们必须明确：任何专利必须以允许每个人自由使用为前提，否则就不准许授予专利。

关于复制、发行和修改的精确条款与条件如下。

GNU 程序通用公共许可证有关复制，分发和修改的条款和条件

0. 本许可证适用于任何这样的程序或其他作品，它的版权拥有者声称它是按照通用公共许可证分发的，并且该声明被包含在其中。下面提到的“程序”指的是任何按照以下条款分发的程序或作品。“基于程序的作品”指的是程序本身或在版权法意义下的任何从程序中派生的作品：就是说，可以是包括程序或者程序的一部分的作品，还可以是逐字逐句地、或者带有修改地、或者直截了当地翻译到其他的语言。（在下文中，任何翻译作品都被包含在条款“修改”中。）每一个获得许可证的人都被称为“你”。

许可证条款不适用于复制，分发和修改以外的活动。它们超出了本许可证条款的范围。运行程序的行为不受本许可证条款的限制。仅仅在程序的输出内容构成了基于该程序的作品的情况下（不依赖为了运行程序而采取的行为），程序的输出才适用于本条款。是否适用取决于程序用来做什么。

1. 如果你在的程序每个拷贝中适度突出地发表适当的版权声明并拒绝提供任何担保，并且保持本许可证的声明和拒绝担保的声明的完整，并且与程序一同分发本许可证，那么你就可以通过任何媒体完整地复制和分发你所得到程序的全部源代码。

你可以对散发物理拷贝收费，你还可以按照你的意愿提供担保保护以作为收费的报偿。

2. 如果你满足了以下所有条件，你就可以对你的拷贝、程序的拷贝或者它的任何一个部分进行修改，从而形成基于程序的作品，并且按照条款 1 复制和分发这项作品。

a）你必须在被修改的文件中突出地声明你对该文件的所做的任何修改和修改的日期。

b）你必须允许你的作品作为整体由任何第三方按照本许可证的条款自由使用。

c）假如被修改的程序运行时交互地读入命令，当交互式运行该程序时，你需要打印或显示包括合适的版权声明和拒绝担保（你也可以提供担保）的通告，并且告诉用户可以在这些条件下重新分发该程序，以及告诉用户如何阅读许可证的拷贝。（例外的情况：假如程序本身是交互式的，但一般并不打印这些通告，则你的基于这个程序的作品不需要打印通告。）

该要求适用于被修改作品的整体。如果可以确认该作品的某个部分不是从程序中派生的，并且可以被合理地认为是独立和分离的作品，那么在你将它作为独立的作品分发时，则不需受本许可证和其中的条款约束。但是在你把它作为基于程序的作品的一部分

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

进行分发的时候，作为整体它必须按照本许可证分发。不论其中各个部分是由谁写的，对于其他许可证获得者来说许可都扩展到整个作品。

因此，本条款的意图不在于索取权利或剥夺完全由你创作的作品的权利；而在于行使控制对派生作品和由集体创作的基于程序的作品的分发权力。

另外，把不基于程序的其他作品和程序或基于程序的作品一起放在同一个存储体或分发媒体的同一卷上，并不能将其他作品也置于本许可证的约束范围之内。

3. 你可以在第 1 款和第 2 款的约束下以目标代码或可执行的形式复制和分发程序（或者在第 2 款的约束下复制和分发程序的一部分或程序的派生作品），但须按如下方式之一：

a）在常用的软件交流媒体上按上面的第 1 款和第 2 款的约束分发机器可读的、与目标代码或可执行形式相对应的完整源代码。

b）在常用的软件交流媒体上按上面的第 1 款和第 2 款的约束书面分发机器可读的、与目标代码或可执行形式相对应的完整源代码的拷贝，该书面分发至少三年有效，并且费用不能超过源代码的实际分发费用。

c）提供你所收到的与目标代码或可执行形式相对应的完整源代码的分发说明。（这种方式仅可用于非商业方式，并且你按照上面的 b）收到程序的目标代码或可执行形式。）

作品的源代码指的是对作品进行修改最优先择取的形式。对于可执行的作品，完整的源代码包括：所有模块的所有源程序、任何相关的接口的定义、控制可执行作品的安装和编译的脚本。作为特殊例外，发布的源代码不必包含任何常规发布的供可执行代码在上面运行的操作系统的主要组成部分（如编译程序、内核等）。除非这些组成部分和可执行作品结合在一起。

如果采用提供对指定地点的访问和复制的方式发布可执行代码或目标代码，那么，提供对同一地点的访问和复制源代码可以算做源代码的发布，即使第三方不强求与目标代码一起复制源代码。

4. 除非你明确按许可证提出的要求去做，否则你不能复制、修改、转发许可证和发布程序。任何试图用其他方式复制、修改、转发许可证和发布程序是无效的。而且将自动结束许可证赋予你的权利。然而，对那些从你那里按许可证条款得到副本和权利的人们，只要他们继续全面履行条款，许可证赋予他们的权利仍然有效。

5. 你没有在许可证上签字，因而你没有必要一定接受这一许可证。然而，没有任何其他东西赋予你修改和发布程序及其衍生作品的权利。如果你不接受许可证，这些行为是被法律禁止的。因此，如果你修改或发布程序（或任何基于程序的作品），你就表明你接受这一许可证以及它的所有有关复制，发布和修改程序或基于程序的作品的条款和条件。

6. 每当你重新发布程序（或任何基于程序的作品）时，接受者自动从原始许可证颁发者那里接收到这些条款和条件支配的复制，发布或修改程序的许可证。你不可以对接受者履行这里赋予他们的权利强加其他限制。你也没有强求第三方履行许可证条款的义务。

7. 如果由于法院判决或违反专利的指控或任何其他原因（不限于专利问题）的结果，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com

强加于你的条件（不管是法院判决，协议或其他）和许可证的条件有冲突。他们也不能用许可证条款为你开脱。在你不能同时满足本许可证规定的义务及其他相关的义务时，作为结果，你可以根本不发布程序。例如，如果某一专利许可证不允许所有那些直接或间接从你那里接受副本的人们在不付专利费的情况下重新发布程序，惟一能同时满足两方面要求的办法是停止发布程序。

如果本条款的任何部分在特定的环境下无效或无法实施，就使用条款的其余部分。并将条款作为整体用于其他环境。

本条款的目的不在于引诱你侵犯专利或其他财产权的要求，或争论这种要求的有效性。本条款的主要目的在于保护自由软件发布系统的完整性。它是通过通用公共许可证的应用来实现的。许多人坚持应用这一系统，已经为通过这一系统发布大量自由软件作出慷慨的贡献。作者／捐献者有权决定他／她是否通过任何其他系统发布软件。许可证持有人不能强制这种选择。

本节的目的在于明确说明许可证其余部分可能产生的结果。

8. 如果由于专利或者由于有版权的接口问题使程序在某些国家的发布和使用受到限止，将此程序置于许可证约束下的原始版权拥有者可以增加限制发布地区的条款，将这些国家明确排除在外。并在这些国家以外的地区发布程序。在这种情况下，许可证包含的限止条款和许可证正文一样有效。

9. 自由软件基金会可能随时出版通用公共许可证的修改版或新版。新版和当前的版本在原则上保持一致，但在提到新问题时或有关事项时，在细节上可能出现差别。

每一版本都有不同的版本号。如果程序指定适用于它的许可证版本号以及“任何更新的版本”。你有权选择遵循指定的版本或自由软件基金会以后出版的新版本，如果程序未指定许可证版本，你可选择自由软件基金会已经出版的任何版本。

10. 如果你愿意将程序的一部分结合到其他自由程序中，而它们的发布条件不同，应写信给作者，要求准予使用。如果是自由软件基金会加以版权保护的软件，写信给自由软件基金会。我们有时会作为例外的情况处理。我们的决定受两个主要目标的指导。这两个主要目标是：我们的自由软件的衍生作品继续保持自由状态。以及从整体上促进软件的共享和重复利用。

没 有 担 保

11. 由于程序准予免费使用，在适用法准许的范围内，对程序没有担保。除非另有书面说明，版权所有者和／或其他提供程序的人们“一样”不提供任何类型的担保。不论是明确的，还是隐含的。包括但不限于隐含的适销和适合特定用途的保证。全部的风险，如程序的质量和性能问题都由你来承担。如果程序出现缺陷，你承担所有必要的服务，修复和改正的费用。

12. 除非适用法或书面协议的要求，在任何情况下，任何版权所有者或任何按许可证条款修改和发布程序的人们都不对你的损失负有任何责任。包括由于使用或不能使用程序引起的任何一般的、特殊的、偶然发生的或重大的损失（包括但不限于数据的损失，

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科 www.zzbaike.com 本教程由站长百科收集整理

或者数据变得不精确，或者你或第三方的持续的损失，或者程序不能和其他程序协调运行等）。即使版权所有者和其他人提到这种损失的可能性也不例外。

最后的条款和条件

如何将这些条款用到你的新程序

如果你开发了新程序，而且你需要它得到公众最大限度的利用。

要做到这一点的最好办法是将它变为自由软件。使得每个人都能在遵守条款的基础上对它进行修改和重新发布。为了做到这一点，给程序附上下列声明。最安全的方式是将它放在每个源程序的开头，以便最有效地传递拒绝担保的信息。每个文件至少应有“版权所有”行以及在什么地方能看到声明全文的说明。

用一行空间给出程序的名称和它用来做什么的简单说明 Copyright ©<年><作者姓名>

这一程序是自由软件，你可以遵照自由软件基金会出版的 GNU 通用公共许可证条款来修改和重新发布这一程序。或者用许可证的第二版，或者（根据你的选择）用任何更新的版本。

发布这一程序的目的是希望它有用，但没有任何担保。甚至没有适合特定目的的隐含的担保。更详细的情况请参阅 GNU 通用公共许可证。

你应该已经和程序一起收到一份 GNU 通用公共许可证的副本。如果还没有，写信给：Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

还应加上如何和你保持联系的信息。

如果程序以交互方式进行工作，当它开始进入交互方式工作时，使它输出类似下面的简短声明：

Gnomovision 第 69 版，版权所有©年、作者姓名，Gnomovision 绝对没有担保。要知道详细情况，请输入“show w”。这是自由软件，欢迎你遵守一定的条件重新发布它，要知道详细情况，请输入“Show c”。

假设的命令“shovr w”和“show c”应显示通用公共许可证的相应条款。当然，你使用的命令名称可以不同于“show w”和“show c”。根据你的程序的具体情况，也可以用菜单或鼠标选项来显示这些条款。

如果需要，你应该取得你的上司（如果你是程序员）或你的学校签署放弃程序版权的声明。下面只是一个例子，你应该改变相应的名称：

Ynyodyne 公司以此方式放弃 James Harker 所写的 Gnomovision 程序的全部版权利益。<Ty coon 签名>，1989.4.1 Ty coon 付总裁。

这一许可证不允许你将程序并入专用程序。如果你的程序是一个子程序库。你可能会认为用库的方式和专用应用程序连接更有用。如果这是你想做的事，使用 GNU 库通用公共许可证代替本许可证。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

附录 D　通用公共许可证的法律要点和热点问题的阐释

自由软件运动在整个软件产业界成了一股影响力越来越大的发展趋势；自由软件产品的使用也越来越广泛；整个自由软件社区呈蓬勃发展之势。公共许可证（GPL）是自由软件运动的重要的法律基石之一。然而，许多人，包括 IT 专业人士，甚至软件专家、开源软件技术爱好者都对 GPL 存在一定程度的误解，这在一定程度上影响了自由软件的发展。本附录将对通用公共许可证的几个主要的要点和热点问题进行阐释。

D.1　GPL 是否与版权法和软件开发者的权利相冲突

首先需要强调的一点是，遵循 GPL 的软件也是有版权的和受许可证限制的。虽然能够免费获取自由软件，但并不是说自由软件放弃了著作权。与商用软件相同，必须遵守使用授权合同。侵犯了 GPL 软件的版权也同样需付法律责任。为了与通常的版权（copyright）相区分，GPL 软件的版权又称为 copyleft。

何谓版权？版权也叫著作权，它是作品的创作者依照法律取得的对作品的专有权利。版权法律制度的建立，是为了保证作者能从自己创作的作品中获得精神上的承认和物质上的利益，从而鼓励创作者的热情，为社会不间断地提供优秀文化作品。版权法律制度虽然客观上给予了作者相当多的专有权利，但仍是以社会的繁荣进步，满足公众利益为第一目的的。因此，版权法条文的设立，必须考虑作者权利和公众利益的平衡。

大多数私有软件的许可证是用来剥夺分享和修改软件的自由，而 GPL 则是保证分享和修改软件的自由。GPL 的本质是以新式版权来替代传统版权。

世界上大多数国家都对计算机软件赋予了版权。然而，软件的确与传统的文字作品具有重大区别，软件有着被作为工具使用的特点。某些软件公司过度利用了版权法赋予的专有权利，让计算机的公众用户承担了昂贵的软件使用费用。在有些情况下，软件所有人的权利与用户的利益明显失衡。

我国将《计算机软件保护条例》作为《著作权法》之下的特别行政法规开始实施，但计算机软件保护制度与一般著作权保护有些差异。例如，我国 1991 年制定的《计算机软件保护条例》规定可在没有得到授权的情况下，对计算机软件合理使用。规定因课堂教学、科学研究、国家机关执行公务等非商业性目的的需要对软件进行少量的复制，可以不经软件著作权人或者其合法受让者的同意，不向其支付报酬，但使用时应当说明该软件的名称、开发者，并且不得侵犯著作权人或者其合法受让者依本条例所享有的其他

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

各项权利。该复制品使用完毕后应当妥善保管、收回或者销毁，不得用于其他目的或者向他人提供。2002 年 1 月 1 日起施行的新颁《计算机软件保护条例》完全取消了原条例中关于“合理使用”的规定，即“因课堂教学、科学研究、国家机关执行公务等非商业性目的的需要，对软件进行少量的复制，可以不经软件著作权人或者其合法受让者的同意，不向其支付报酬”，从而把我国的一切单位和个人，在软件未经授权情况下的合理使用，统统定为非法，并要加以处罚。2002 年 10 月 15 日，最高人民法院施行了《关于审理著作权民事纠纷案件适用法律若干问题的解释》。其中，第二十一条规定：“计算机软件用户未经许可或者超过许可范围商业使用计算机软件的，依据著作权法第四十七条第（一）项、《计算机软件保护条例》第二十四条第（一）项的规定承担民事责任。”即用户在未经许可使用软件的侵犯民事责任，限制在“商业使用”的范围内。

我国 1991 年制定的《计算机软件保护条例》规定因下列情况之一而引起的所开发的软件与已经存在的软件相似，不构成对已经存在的软件的著作权的侵犯：

（一）由于必须执行国家有关政策、法律、法规和规章；

（二）由于必须执行国家技术标准；

（三）由于可供选用的表现形式种类有限。

由此可以看出，计算机软件原先在中国著作权法之下受保护的水平明显低于一般文学艺术作品。2002 年 1 月 1 日，修订后的《计算机软件保护条例》开始生效。在新条例之下，计算机软件享受的保护不仅不再低于一般文学艺术作品，而且在某些方面又显示出高于一般文学艺术作品的趋势，但依然规定，为了学习和研究软件内含的设计思想和原理，通过安装、显示、传输或者存储软件等方式使用软件的，可以不经软件著作权人许可，不向其支付报酬。

软件业用版权法来保护软件开发企业和个人的权利，用许可证来约束这些权利的行使和转移。传统的版权法保护软件所有者使用、销售、修改和衍生软件的权利，这些权利可以通过许可证转让给其他方。

近年来，包括一些欧、美学者在内，越来越多的人认识到，知识产权保护的最终目的应当是形成一种公平、合理的社会秩序，因而知识产权法实际上是一种在特定环境下公平、合理地分配社会利益的规则。既然是分配，必然会涉及多方面的利益分享者。在这种情况下，过分强调任何一个利益群体的利益都是不合适的。因此，将合理使用视为社会公众的权利更合适。当然，此种权利也必须经过谨慎而合理的设计，并根据一个国家的具体情况随时加以调整。

著作权是法律赋予著作权人的权利；合理使用则是法律赋予著作权人以外的其他社会公众的权利。因此，不能单方面强调保护一个方面的权利而忽视另一个方面的权利。GPL 放弃了传统意义上的版权所有者对复制权、发行权和修改权的控制，任何人无须申请许可，就可自由地使用、修改和传播 GPL 软件。然而 GPL 只是限制了直接销售软件获利，并没有完全没有考虑到软件开发者 / 组织的利益。

GPL 并不排斥从软件中获益。任何软件企业和开发者都需要有经济来源才能生存，否则将难以为继，更谈不上发展了。遵循 GPL 的软件开发者仍能通过多种商业模式从 GPL 软件中获益。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

GPL 软件的开发者可以利用自由软件提高品牌知名度和美誉度，建立市场领导者的地位，从而推动其私有软件的销售。例如 IBM 公司大力支持 Eclipse 项目，投入了大量的资金，但由于其私有软件 WSAD 即建立在 Eclipse 技术基础之上，Eclipse 社区的发展培养了大量的 WSAD 开发者和支持者，从而极大地推动了 WSAD 的普及和销售。

对于含有硬件产品的公司，可以利用自由软件推动硬件产品的销售。如 IBM 大力支持 Linux 操作系统，Linux 可以运行在多个系列的硬件平台上，反过来，这又大大促进了 IBM 硬件产品的销售。

对于 GPL 软件，虽然不能直接通过销售许可权获益，但软件开发企业和开发人员可以利用和软件相关的服务和技术咨询等获益。GPL 软件的开发企业在争取这类服务项目时，显然将比其竞争对手具有更大的品牌优势和技术优势。例如，IBM 身为全世界拥有最多专利的公司，但积极支持 GPL 授权方式和自由软件产品。

由于软件开发组织和个人能有多种方式从自由软件中获益，因此大量的自由软件公司不仅维持了生存，而且在商业领域大展身手，呈快速发展之势。

D.2 使用遵循通用 GPL 的软件是否会导致相关的私有软件的强制开源

GPL 规定程序的一部分若是遵循 GPL 进行分发的作品，作为整体，该程序也必须按照 GPL 分发。不论其中各个部分是由谁写的，对于其他许可证获得者来说许可都扩展到了整个作品。当非 GPL 代码与 GPL 代码混合使用时，由此产生的软件就要根据 GPL 来处理。因此许多企业在考虑使用自由软件，都对 GPL 的使用授权等有所顾虑，害怕与自由软件组合使用的私有软件不得不开放其源代码，有关非 GPL 知识产权的权利和利益都不复存在。这样一来，企业面临的风险就会骤然增大，

某著名软件公司的 CEO 亦指称遵循 GPL 条款的自由软件是“病毒”，意指由于使用自由软件，企业将不得不将相关的私有软件公开源代码，并由此形成“雪球”效应，使得越来越多的私有软件被强制公开其源代码，从而使得软件产业失去版权的保护。

其实这种对于 GPL 的“传染性”的担忧是不必要的。虽然 GPL 规定基于 GPL 条款的软件也必须遵循 GPL，然而在英文版的正式的 GPL 条款中，GPL 并没有使用“组合（combination）”之类的词，而是使用了“包含或衍生自（contains or is derived from）”来表达一个特定的软件是否基于 GPL 软件。这些术语在版权法中有非常明确的含义，例如“衍生作品”的定义为：

“基于一个或多个已存在的作品而形成的作品，如翻译、音乐编曲、戏剧改编、小说改编、动画制作、录音合成、艺术再现、删节、缩写，或以其他形式对作品进行了修改、变换或调整。一件包含编辑校对、注释、详解，或是其他修改的作品，如果从整体上是一个有版权的原创作品，就称为衍生作品”。

如果只是将私有软件与自由软件简单地一起使用，并不会构成衍生作品，因为直接将软件合在一起并没有构成一个有版权的原创作品。GPL 条款中的精确的表述可以很明

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

确地避免 GPL 向许可证接受人自己的私有软件延伸。

如果只是简单地使用自由软件，而不对自由软件进行修改，一般说来，就不会产生衍生作品，相应地，相关的私有软件也无需强制遵循 GPL。

对于包含 GPL 软件的混合软件，如果软件版权的所有人不希望保持软件的私有性，则问题要复杂得多。从技术上讲，是否要强制开放整个混合软件得源代码，将主要取决于混合软件中的 GPL 软件和混合软件中的其他软件的链接关系。

对于在混合软件中通过静态链接的形式链接到遵循 GPL 的软件，由于静态链接需要修改链接程序的代码，这样将形成了衍生作品，从而衍生作品也需要遵循 GPL。因此对于在软件中用到自由软件，而又不希望开放新作品的源代码的开发者来说，不能使用静态链接的形式使用自由软件，而应该使用动态链接。

动态链接只是在程序间设立瞬间性联系，无需修改链接程序，从而将不会形成衍生产品，因此无需将以动态链接形式链接到 GPL 软件的私有软件强制其遵循 GPL。

此外，对于利用应用程序编程接口（API）进行私有程序和 GPL 间的数据交换的混合程序来说，如果只是进行简单的数据交换，即使这些数据会影响到 GPL 程序的行为，也不会形成衍生作品，私有软件也无需强制遵循 GPL。

此外，开发者完全可以借鉴自由软件的经验和思想开发私有软件，无需担心侵权问题。虽然基于 GPL 软件的修改后的衍生作品仍需遵循 GPL，但版权法保护的是作品的表达而非思想。所以自由软件对于私有软件的开发仍有很大的借鉴意义。

D.3 使用自由软件是否会给软件使用者带来巨大的风险

和私有软件类似，自由软件也涉及侵犯专利权问题。例如，MIT 的自由软件 X Windows 系统遭到了 AT&T 公司的专利侵权诉讼，虽然相关的技术 MIT 甚至先于 AT&T 使用，但是，由于 MIT 的软件人员认为这项技术太普通了，而没有公开发表。结果，AT&T 的专利申请被专利机构批准，MIT 不能再使用该专利技术。

对于遵循 GPL 的自由软件来说，AT&T 诉 MIT 的 X Windows 系统之类的专利权案一般不会发生。因为 GPL 条款中已经制约了自由软件的再发布者以私有名义将自由软件中的某项技术获得专利。GPL 明确地规定：要么申请到专利后允许任何人自由使用（这等同于没有专利），要么不准许有专利，从而防止了将自由软件技术私有化的倾向。

然而，自由软件可能存在这样一类专利侵权案：自由软件的原始开发者或后续修改者在程序或其衍生作品上使用了不受 GPL 约束的专利技术。对于这类情况，GPL 规定：如果某一专利不允许所有那些直接或间接获得程序副本的人们在不付专利费的情况下再发布程序，则不应发布相关程序。

SCO 曾指控 IBM 将 UNIX 技术用在 Linux，违反了 SCO 所允许的 UNIX 授权范围。SCO 发函给全球 1500 家大企业，警告称，使用 Linux 可能触犯法律。

在使用遵循 GPL 条款的自由软件时，有些企业可能会存在以下担心：在企业中使用了自由软件，并根据具体需求进行了修改，是否需要被强制公开其源代码，从而暴露了企业秘密和私有信息。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

对于以上担忧，完全没有必要。GPL 只是规定重新发布后的软件及其衍生作品也许遵循 GPL，而对于私人或企业内部中的使用及其修改、加上无需公开源代码。之所以无需公开源码，是因为不满足源码公开义务的条件，公司内部使用不属于重新发布。但是，在向相关公司及分公司提供程序时，如果作为集成商构建了用户企业的系统，那么将被视为重新发布了程序，这时必须公开源码。

GPL 本身的合法性的问题，还存有一些争议。在与 IBM 的官司纠纷里，SCO 开始直接挑战 GPL。SCO 称："GPL 违背美国宪法、版权法、反托拉斯法，及出口管制等法令"。此外，SCO 还宣称，GPL 是无效的。其焦点争执在于"授权"。而 IBM 及大多数法律界人士则表示了对 GPL 的信心。

SCO 与 IBM 关于 Linux 的侵权官司的事件，IBM 认为，既然 SCO 先前作为一个 Linux 厂商（现在 SCO 实际上是以前的 Linux 厂商 Caldera）基于 GPL 发布了包括 Linux 核心源代码在内的 Linux 系统，其中包括了这次 SCO 认为所谓"侵权"的部分，由于这种许可证的发放是不可撤销的，因此 SCO 不再享有对这些代码的权利。

虽然 GPL 许可证的形式已获得美国政府版权受理机构的正式批准（专利号： 59 Temple Place –Suite 330, Boston, MA 02111-1307, USA），因此受到法律保护，但到目前为止，GPL 尚未经受严峻的法律事件的挑战。SCO 对 IBM 的诉讼给 GPL 提供了一个独特的机会——为 GPL 获得一些法律上的确认。

令自由软件爱好者释然的是，根据德国慕尼黑地区法院的类似案例的判决，法院认为 GNU GPL 条款的主要内容符合德国版权以及合同法的相关规定。这也是在世界范围内，GNU GPL 首次得到法院的认可。法院用了 20 页的篇幅说明了 GPL 合法性的根据。目前宣判结果以及相关文件的德文版已经问世，英文版也即将公布。由于目前发生 SCO 质疑 GPL 之类的事件，这样的判决结果具有重大意义。

对于自由软件给使用者所带来的风险问题，许多法律机构和大型组织的调查和分析结论是："如果是自家公司内部使用，则关于专利侵害等的风险并不高"。

GPL 许可证可能存在的问题

GPL 许可证没有规定许可证的永久性。这个问题可能在涉及自由软件项目的时，留下许可证"合理期限"已过的隐患。

由于 GPL 以美国的著作权法为基础，因此存在不符合其他国家法律的可能，例如我国的著作权法就与美国有所不同。虽然 GPL 规定：若 GPL 条款和有些国家、地区的法律相冲突，发布遵循 GPL 软件的范围可将这些国家排除在外，但这可能会给这些国家的使用者和重新发布者带来一些潜在的问题。此外，当遵循 GPL 软件的争端涉及跨国时，如何处置，例如，将适用何国法律等，这些都是问题。

从上述讨论可以看出，我们在自由软件运动中需要注意以下几点：

- 应当学会遵守自由软件运动的游戏规则。
- 无论是使用者还是开发者都必须注意到自由软件与私有软件不同。
- 商业协议的条文必须十分精确、明确，以避免可能产生的不同解释。

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

参考文献

1. 工作台用户指南. Eclipse2.1 联机帮助
2. Java 开发用户指南. Eclipse2.1 联机帮助
3. 平台插件开发者指南. Eclipse2.1 联机帮助
4. JDT 插件开发者指南. Eclipse2.1 联机帮助
5. PDE 指南. Eclipse2.1 联机帮助
6. Object Technology International，Inc. Eclipse Platform Technical Overview. http://www.eclipse.org
7. http://www.eclipse.org
8. http://www.opensource.org
9. http://yangxingjun.myrice.com/chinesehow/kernel/node33.html
10. lily. http://www.20cn.net/ns/cn/zs/data/20010129033504.htm
11. http://www.gnu.org/licenses/gpl.txt
12. 王立. 关于开放源代码的定义. http://master.linux.net.cn/~charles/translations/osi.gb2312.php
13. 王立. GNU 库通用公共许可证 http://master.linux.net.cn/~charles/translations/libgpl.gb2312.php
14. 贾星客，李极光. 论自由软件运动.《云南师范大学学报》第 33 卷第 1 期，2001 年 1 月
15. 共创软件联盟. 关于开放源代码和通用公共许可（GPL）的进一步讨论——共创软件联盟提交科技部的调研报告. 共创软件. 2002，(6).-3-14
16. 江何，周华春. 开放源代码软件的应用研究. 计算机系统应用. 2002，(6).-77-79
17. 董颖，胡君玲. 自由软件——自由软件、开放源代码和 GPL. http://www.penguin64.com.cn/freeware/2000414-3-1.htm
18. 董向辉. 在 Eclipse 2.0 中使用版本控制系统 CVS. http://www-900.ibm.com/developerWorks/cn/java/l-cvs/index.shtml#author1
19. 杨锦方. CVS 和 Night Build 技术. 清华大学出版社. 2002 年 10 月第 1 版
20. 倪大鹏. Eclipse 入门——Eclipse 的使用简介及插件开发 http://www-900. ibm.com/developerWorks/cn/java/l-eclipse/index.shtml
21. 倪大鹏. SWT：AWT 和 SWING 的强大竞争者 http://www-900.ibm.com/developerWorks/cn/java/l-swt/index.shtml
22. 丁鹏，刘方，邵志雄，何柄胜编著. STR UTS 技术揭密及 WEB 开发实例. 清华大学出版社，2004 年 3 月第 1 版

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

23. Gary Craig, Peter Jakab 著，张云涛，龚玲，张燕译. IBM WebSphere Studio Web 应用开发. 机械工业出版社，2004 年 2 月第 1 版
24. 刘冬. 在 Eclipse 中开发 struts 应用程序 http://www-900.ibm.com/developerWorks/cn/java/1-eclipse-struts/
25. Nancy Chen Junhua. 使用 Easy Struts for Eclipse 开发 Struts. http://www-900.ibm.com/developerWorks/cn/linux
26. Clifford Spinac. 创建基于 Struts 的 Web 应用程序. developerWorks 专刊，2003 年第 1 期
27. Adian Van Emmenis. Using the Eclipse GUI outside the Eclipse Workbench, Part 1: Using JFace and SWT in stand-alone mode http://www-900.ibm.com/developerWorks/cn/linux/opensource/os-ecgui1/index_eng.shtml
28. 董向辉. VAJ 之后是什么 http://www-900.ibm.com/developerWorks/cn/java/ l-ide/part2/index.shtml
29. 倪大鹏. Eclipse 入门——Eclipse 的使用简介及插件开发 http://www-900.ibm.com/ developerWorks/cn/java/l-eclipse/index.shtml
30. 陈际红. 从法律角度解读 Linux 商业化问题及在中国前景 http://it.sohu.com/2004/04/19/35/article219883512.shtml
31. 唐广良. 计算机软件保护制度存在的问题及其解决 http://www.iolaw.org.cn/shownews.asp?id = 4084
32. Geoffrey R. Duck.使用 Eclipse 作为 Jakarta Tomcat 的开发环境—— 一种快速集成 Eclipse 和 Tomcat 的方法. http://www-900.ibm.com/developerworks/cn/opensource/os-ectom/
33. 左锦. 使用 Eclipse 开发 JSp http://www-900.ibm.com/developerWorks/cn/java/l-jsp-eclipse/index.shtml
34. 孙卫琴，李洪成. Tomcat 与 Java Web 开发技术详解. 电子工业出版社，2004 年 4 月第 1 版
35. Ashton 著，丁一清，吴静译. 自学 JDBC 数据库编程. 清华大学出版社，1998 年 8 月第 1 版
36. Jason Brittain，Ian F. Darwin 著，陈光译. Tomcat 配置技巧 Top 10. http://www.oreilly.com.cn/news.php?n=20040109171925&c=java
37. 马永征. 单元测试. http://ntarl.cnic.ac.cn/html/ppt_down/test-myz20040615.ppt
38. 网辉. junit 使用简明手册. http://hk109.yculblog.com/post-46260.html
39. IBM. wf3115inst.pdf. IBM Learning Services Worldwide Certified Material
40. Ueli Wahli 等著，瞿裕忠，张剑峰，王从刚等译. WebSphere 环境下 Servlet 与 JSP 高级编程. 机械工业出版社，2001 年 8 月第 1 版
41. Howard Kushner 主编，张云涛，龚玲，马颖华，苏贵洋译. IBM WebSphere Studio J2EE 应用开发. 机械工业出版社，2004 年 4 月第 1 版

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

42. 李巍. JSP 编程入门与应用实例. 清华大学出版社，2001 年 1 月第 1 版
43. Heavyz. Tomcat 启动分析. http://www.huihoo.org/apache/tomcat/heavyz/01-startup.html
44. 王克宏主编. Java 2 程序设计. 清华大学出版社，2000 年 9 月第 1 版
45. Koray Guclu 著，skyhit 译. 使用 Eclipse 进行 SWT 编程 http://www.matrix.org.cn/blog/skyhit/archives/000586.html
46. Malcolm Davis. Struts，MVC 的一种开放源码实现 http://www-900.ibm.com/developerWorks/cn/java/j-struts/index.shtml
47. 赵晨希. 用Struts建立MVC应用的介绍http://www-900.ibm.com/developerWorks/cn/java/l-struts-mvc/index.shtml#author1
48. Cbuck Cavaness 著. 孙江宏，杨继萍译. Jakarta Struts 编程. 清华大学出版社，2004 年 4 月第 1 版
49. 王和全. 深入 Struts 1.1. http://www-900.ibm.com/developerWorks/cn/java/l-struts1-1/
50. 务实. 浅析 Struts 体系结构与工作原理 http://www.yesky.com/SoftChannel/72342371961929728/20040315/1777062_1.shtml
51. chensheng913. 选择合适的 JDBC 驱动程序. http://dev.csdn.net/article/33/33607.shtm
52. 陈光. Tomcat4 的数据库连接池配置 http://www.zdnet.com.cn/developer/code/story/0,2000081534,39134180-1,00.htm
53. Eric. Ant 实践. http://www.huihoo.com/java/ant.html
54. Eric M. Burke 著，stonexu 译. ANT 十五大最佳实践（上）http://tech.ccidnet.com/pub/article/c1077_a125195_p1.html
55. Jesse Tilly, Eric M. Burke 著，林琪译. Ant 权威指南. 中国电力出版社，2003 年 8 月第 1 版
56. Pawel Leszek. Eclipse 如何使用源代码版本控制 http://www-900.ibm.com/developerWorks/cn/linux/opensource/os-ecshare/index.shtml

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。

站长百科
www.zzbaike.com
本教程由站长百科收集整理

42. 李巍. JSP 编程入门与应用实例. 清华大学出版社，2001 年 1 月第 1 版
43. Heavyz. Tomcat 启动分析. http://www.huihoo.org/apache/tomcat/heavyz/01-startup.html
44. 王克宏主编. Java 2 程序设计. 清华大学出版社，2000 年 9 月第 1 版
45. Koray Guclu 著，skyhit 译. 使用 Eclipse 进行 SWT 编程 http://www.matrix.org.cn/blog/skyhit/archives/000586.html
46. Malcolm Davis. Struts，MVC 的一种开放源码实现 http://www-900.ibm.com/developerWorks/cn/java/j-struts/index.shtml
47. 赵晨希. 用Struts建立MVC应用的介绍http://www-900.ibm.com/developerWorks/cn/java/l-struts-mvc/index.shtml#author1
48. Cbuck Cavaness 著. 孙江宏，杨继萍译. Jakarta Struts 编程. 清华大学出版社，2004 年 4 月第 1 版
49. 王和全. 深入 Struts 1.1. http://www-900.ibm.com/developerWorks/cn/java/l-struts1-1/
50. 务实. 浅析 Struts 体系结构与工作原理 http://www.yesky.com/SoftChannel/72342371961929728/20040315/1777062_1.shtml
51. chensheng913. 选择合适的 JDBC 驱动程序. http://dev.csdn.net/article/33/33607.shtm
52. 陈光. Tomcat4 的数据库连接池配置 http://www.zdnet.com.cn/developer/code/story/0,2000081534,39134180-1,00.htm
53. Eric. Ant 实践. http://www.huihoo.com/java/ant.html
54. Eric M. Burke 著，stonexu 译. ANT 十五大最佳实践（上）http://tech.ccidnet.com/pub/article/c1077_a125195_p1.html
55. Jesse Tilly, Eric M. Burke 著，林琪译. Ant 权威指南. 中国电力出版社，2003 年 8 月第 1 版
56. Pawel Leszek. Eclipse 如何使用源代码版本控制 http://www-900.ibm.com/developerWorks/cn/linux/opensource/os-ecshare/index.shtml

更多电子书教程下载请登陆http://down.zzbaike.com/ebook
本站提供的电子书教程均为网上搜集，如果该教程涉及或侵害到您的版权请联系我们。